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 реалізації інтелектуального DLP-агента”

Виконав здобувач вищої освіти
IV курсу, групи КН-22
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Кіхоть М.В.
«____» _____ 2026 р.

Керівник проекту
кандидат технічних наук
_____ Лисенко І.А.
«____» _____ 2026 р.
Рецензент _____
_____

м. Кропивницький

Центральноукраїнський національний технічний університет
Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Освітній ступінь *бакалавр*
Галузь знань_12 _____*"Інформаційні технології"*_____
Спеціальність 122 *"Комп'ютерні науки"*
Освітньо-професійна (освітньо-наукова) програма _*"Комп'ютерні науки"*_

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

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

Кіхотю Максиму Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи реалізації інтелектуального DLP-агента*

2. Керівник роботи *Лисенко Ірина Анатоліївна, канд. техн. наук*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 116-02 від 06.02.2026 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 реалізації інтелектуального DLP-агента*

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 « 6 » 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
« 6 » лютого 2026 р.

Підпис керівника
_____ Лисенко І.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 6 » лютого 2026 р.

Підпис здобувача
_____ Кіхоть М.В.
(прізвище та ініціали)

АНОТАЦІЯ

Кіхоть М.В. Програмне забезпечення системи реалізації інтелектуального DLP-агента. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи реалізації інтелектуального DLP-агента.

Метою розробки є програмне забезпечення системи реалізації інтелектуального DLP-агента.

Результат роботи – програмна реалізація системи реалізації інтелектуального DLP-агента.

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 Python.

Ключові слова: комп'ютерні науки, інтелектуальний DLP-агент

ABSTRACT

Kikhot M.V. Software for the implementation of an intelligent DLP agent system. 122 Computer Science. 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 implementation of an intelligent DLP agent system.

The purpose of the development is software for the implementation of an intelligent DLP agent system.

The result of the work is software implementation of the intelligent DLP agent system.

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 OS Windows 10/11.

The program was developed in the Python environment.

Keywords: computer science, intelligent DLP agent

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ДПФ	–	дискретне перетворення Фур'є
ДПХ	–	дискретне перетворення Хартлі
ЕОМ	–	електрона обчислювальна машина
НСД	–	несанкціонований доступ
ПЗ	–	програмне забезпечення
СУБД	–	система управління базами даних
СВВ	–	система виявлення вторгнень
ШПФ	–	швидке перетворення Фур'є
ШПХ	–	швидке перетворення Хартлі
DLP	–	Data Leak Prevention, запобігання витоків даних

ВСТУП

Актуальність теми. Інструменти запобігання втраті даних (DLP) є важливими для організацій, які хочуть захистити конфіденційні дані, зменшити ризики для внутрішніх користувачів та дотримуватися таких норм, як GDPR, HIPAA та SOC 2. Незалежно від того, чи ви керуєте малим бізнесом, чи великим підприємством, вибір правильного рішення DLP допоможе вам захистити кінцеві точки, електронну пошту, веб-трафік та хмарні додатки.

Запобігання втраті даних (DLP) – це набір технологій, які відстежують, виявляють та запобігають витоку конфіденційних даних з вашої організації. Сучасні рішення DLP поєднують засоби контролю кінцевих точок, мережі та хмарних технологій для захисту як від зовнішніх зловмисників, так і від внутрішніх загроз.

Типові випадки використання DLP:

– Відповідність – забезпечення дотримання вимог GDPR, HIPAA, SOC 2.

– Моніторинг внутрішніх загроз – виявлення ризикованої діяльності співробітників.

– Хмарна безпека – захист SaaS-додатків (Microsoft 365, Google Диск, Slack).

– Контроль USB та пристроїв – блокування несанкціонованої передачі даних.

– Класифікація даних та звітність – виявлення та класифікація конфіденційної інформації.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи реалізації інтелектуального DLP-агента.

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

– Огляд існуючих систем реалізації інтелектуального DLP-агента.

– Дослідження системи реалізації інтелектуального DLP-агента.

– Програмна реалізація системи реалізації інтелектуального DLP-агента.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі реалізації інтелектуального DLP-агента.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи реалізації інтелектуального DLP-агента, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Вибираючи DLP-платформу, враховуйте такі критерії:

– Модель розгортання: хмарна, локальна або гібридна.

– Класифікація та автоматизація за допомогою штучного інтелекту: виявлення конфіденційних даних за допомогою машинного навчання та відстеження походження даних.

– Інтеграція: Підтримка кінцевих точок, електронної пошти, SaaS та контролю веб-трафіку.

– Ціноутворення та масштабованість: Доступна ціна для малого та середнього бізнесу порівняно з повноцінними платформами корпоративного рівня.

– Простота використання: просте налаштування для малого бізнесу проти розширеного управління політиками для великих корпорацій.

1.2 Область застосування

Може здатися, що в 2026 році DLP-системи стали настільки звичним явищем в інформаційній безпеці, що питання перехоплення трафіку пішли на другий план. Підходи різних виробників DLP до тому, який трафік і де варто збирати, як правило, ідуть коріннями в їхній власний досвід перших удалих проектів. Очевидно, що досвід цей варіювався від вендора до вендору, оскільки інфраструктура підприємства – річ досить індивідуальна, та й особисті переваги фахівців з інформаційної безпеки теж робили свій вплив.

Проте, у підсумку сьогодні практично будь-яка DLP-система здатна збирати трафік на поштових серверах, мережному шлюзі, проксі-сервері й

кінцевих робочих станціях. Однак цей останній пункт дотепер викликає багато суперечок. Для один агент DLP – панацея від всіх погроз, для інших – страждання. Про цьому й хочеться поговорити в даної розділі.

По класиці DLP, агент потрібно, щоб забезпечувати на робочих станціях контроль даних у спокої (Data-at-rest) і даних у використанні (Data-in-use). Data-in-motion на 95% покривається на рівні серверів, що не може не радувати.

У теорії додавання агентського DLP дозволяє не тільки здійснювати пошук конфіденційної інформації на ПК співробітників, але й контролювати її обіг у ситуаціях, коли машина перебуває не в мережі, або коли інформацією активно користуються. Однак на практиці разом із широкими можливостями агенти несуть із собою й масу труднощів.

Проблеми з установкою

По-перше, агенти дуже проблематично розвертати. З тут криється на пілотних проектах: звичайно агенти без проблем установлюються на кілька десятків або сотень машин, а от коли наступає час промислового впровадження, і тисячі машин повинні обзавестися агентськими модулями, починається саме цікаве. Коли в організації виникають проблеми з установкою DLP-агентів, першим свою дозу негатива, звичайно, одержує конкретний виробник. Але якщо поспілкуватися з досвідченими ІБ-фахівцями, виявляється, що проблема глобальна. Спробуйте пошукати щось начебто «dlp agent not install» або «dlp agent problem». Ви виявите сотні сторінок форумів, де користувачі скаржаться на агентські модулі світових гігантів DLP.

Конфлікти з наявним ПЗ

Були часи, коли будь-який пристойний антивірус моментально детектував DLP-агента як шкідливе ПЗ. Згодом більшості виробників удалося вирішити цю проблему. Але важливо пам'ятати, що ніж могутніший агент і ніж більше схованих функцій ОС він використовує, тим вище ймовірність конфлікту.

По ринку якийсь час ходила історія про якусь організацію в Республіці Білорусь, де був установлений антивірус Вірусблокада. Щоб розгорнути в

організації агентську частину DLP, вендору довелося в терміновому порядку розробляти «інтеграційне рішення» з даним антивірусним ПЗ. Тому коли ви чуєте, що в рішення є інтеграція з тим або іншим антивірусом, у більшості випадків мається на увазі не взаємний обмін подіями ІБ, а саме неконфліктність із DLP-агентом.

Контентний аналіз на робочій станції

Агентський модуль змушений задовольнятися обмеженими ресурсами робочої станції співробітника, а вона звичайно являє собою офісний десктоп або ноутбук, розрахований на роботу з документами, поштовий клієнтом і браузером. При всім при цьому агент повинен здійснювати контентний аналіз прямо на робочій станції. Це б'є по продуктивності, і особливо сильно – якщо організація використовує «крислаті» політики, націлені на захист більших обсягів конфіденційних даних, таких як вивантаження з баз даних або графічних документів. Ви знаєте, як працює OCR на робочій станції або детектор печаток на великому обсязі? Коштує один раз побачити, щоб назавжди запам'ятати.

Помилкові спрацьовування

Останнім часом помітна тенденція до комбінованого використання різних DLP-систем в одній інфраструктурі. Замовники збирають кращі функції від декількох рішень, і потім починається творчий процес налаштування цього зоопарку. Проблема полягає в тому, що різні рішення починають перехоплювати той самий трафік і слати по 2-3 повідомлення на те саме подія – звичайно, у різні консолі. Зрозуміло, це приводить до того, що місце в архівах дуже швидко закінчується. І нарешті, коли робоча станція падає, починається розслідування, який агент у цьому винуватий.

Агенти підтримують не всі платформи

Проблема полягає в тому, що далеко не всі виробники можуть гарантувати стабільну роботу агента із Windows 10/11. Наприклад, якщо зайти на офіційні вітки співтовариств підтримки західних вендорів, можна побачити велику кількість скарг на несумісність із новим утвором Microsoft. Якщо згадати, як

компанія майстерно викручує руки антивірусним виробникам, із продуктами яких ОС всі частіше конфліктує, то складається враження, що Microsoft це на руку й незабаром вони доберуться до інших endpoint-рішень, щоб нав'язати свої умови співробітництва. Користувачам така нестабільність із підтримкою загрожує постійними відновленнями й раптовими відмовами DLP-агентів.

Мобільний агент

Приблизно в 2012 році компанія Symantec ознаменувала нову еру розвитку DLP – запобігання витоків на мобільних пристроях. Їхнє рішення полягало в тому, що iPhone або iPad налаштовано на роботу через VPN-тунель, що направляє весь трафік пристрою на сервер DLP, де відбувається перевірка політик. Що ж, технічно все реалізовано дуже прозоро. Тільки ви коли-небудь пробували проходити цілий день із піднятим VPN? Залишається тільки догадуватися, як швидко розряджається батарея пристрою. Тим більше, у даному прикладі ні про якому агентський DLP на мобільному пристрої мови не йде. Знаючи, як різко негативно Apple відноситься до будь-яких фонових додатків, і, уже тим більше, до тих, які втручаються в процес обробки даних, такому агентові не призначено з'явитися на світло. Невідомо, чому Symantec не розробив DLP-рішення під Android. Можливо, проблема в сегментації операційної системи, і розроблювачам буде дуже складно підтримувати різні версії агентів для всіх моделей телефонів. Та й для більшості процедур перехоплення трафіку буде потрібно root-увати пристрій, а далеко не кожний досвідчений безпечник дозволить робити це навіть довіреному виробникові.

Висновки

Повномасштабне впровадження DLP-системи складно представити без моніторингу кінцевих точок. Обмеження на використання USB-носіїв, контроль буфера обміну – ці канали виходить контролювати тільки на стороні агента.

Якщо ж говорити про комунікаційний трафік, то сучасні технології дозволяють перехоплювати майже все на мережному шлюзі й проксі-сервері. Навіть аналізувати SSL-трафік у прозорому режимі без налаштувань у

властивостях браузера. Тому ми рекомендуємо завжди розумно підходити до вибору перехоплювачів. Занадто часте рішення «простіше на агенті» виявляється глибоко помилковим.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи реалізації інтелектуального DLP-агента, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

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

Kickidler

Kickidler – це рішення для запобігання втратам даних та моніторингу внутрішніх загроз, створене для малого та середнього бізнесу, а також віддалених команд. На відміну від традиційних інструментів, призначених для підприємств, воно зосереджене на видимості кінцевих точок та аналітиці активності користувачів. Kickidler поєднує запис екрана, реєстрацію натискань клавіш та моніторинг продуктивності, щоб виявляти ризиковану поведінку з боку інсайдерів та запобігати випадковим або навмисним витокам даних. Завдяки можливостям локального та хмарного розгортання, Kickidler простий у впровадженні та економічно ефективний, що робить його привабливим для малого та середнього бізнесу, яким потрібно збалансувати захист даних з доступністю. Його функції також підтримують відповідність GDPR та HIPAA, водночас забезпечуючи цінність, що виходить за рамки безпеки, шляхом підвищення ефективності робочої сили.

Переваги: доступна ціна, швидке розгортання, надійне виявлення внутрішніх ризиків, аналітика продуктивності.

Недоліки: Обмежена інтеграція з підприємствами, слабше покриття SaaS/хмарних технологій.

Data Loss Prevention Proofpoint Enterprise

Proofpoint пропонує одну з найбільш визнаних платформ DLP корпоративного рівня, призначену для захисту електронної пошти, веб-трафіку та хмарних додатків. Вона інтегрує класифікацію на основі штучного інтелекту з

розширеною аналітикою загроз для виявлення переміщення конфіденційних даних та внутрішніх ризиків у режимі реального часу. Рішення особливо потужне в DLP електронної пошти, надаючи детальні політики та автоматизовані робочі процеси, які допомагають підприємствам дотримуватися GDPR, HIPAA та SOC 2. Завдяки широкому набору інтеграцій, Proofpoint ідеально підходить для організацій, які значною мірою залежать від екосистем електронної пошти та SaaS, але вона має вищу ціну та вимагає кваліфікованого адміністрування.

Переваги: лідер ринку в сфері захисту від втрати даних електронною поштою, широке охоплення відповідності вимогам, сильна інтеграція SaaS.

Недоліки: Дорого, крута крива навчання, ресурсоємне розгортання.

Data Loss Prevention Symantec (Broadcom)

Symantec DLP, що зараз входить до складу Broadcom, залишається одним із найкомплексніших корпоративних DLP-рішень, що охоплює кінцеві точки, мережі, сховища та хмарні середовища. Він підтримує відстеження походження даних, глибоку перевірку контенту та налаштоване управління політиками. Це робить його кращим вибором для великих підприємств та регульованих секторів, таких як фінанси та охорона здоров'я, де дотримання GDPR, HIPAA та SOC 2 є обов'язковим. Його гібридні варіанти розгортання масштабуються до глобальних організацій, але впровадження може бути складним та ресурсомістким, що вимагає спеціалізованих команд для управління та обслуговування.

Переваги: Повне корпоративне покриття, потужні функції відповідності, масштабоване розгортання.

Недоліки: Висока вартість, складне адміністрування, вимагає спеціалістів.

Data Loss Prevention Forcepoint

Forcepoint робить акцент на управлінні внутрішніми ризиками за допомогою UEBA (аналітики поведінки користувачів та об'єктів) та адаптивного застосування політик. Він забезпечує захист даних на кінцевих пристроях, каналах електронної пошти та веб-трафіку, що робить його універсальним вибором для організацій, що стикаються з внутрішніми загрозами. Підхід

Forcepoint зосереджений на аналізі намірів, зменшенні кількості хибнопозитивних результатів порівняно з системами, що базуються виключно на контенті. Широко використовується в уряді та обороні, він забезпечує дотримання GDPR, HIPAA та SOC 2, зберігаючи при цьому продуктивність користувачів. Однак, точне налаштування є важливим для зменшення шуму та забезпечення точності.

Переваги: Надійне виявлення інсайдерських ризиків, адаптивні політики, що базуються на поведінці, підтримка відповідності вимогам.

Недоліки: Може генерувати хибнопозитивні результати, вимагає постійного налаштування.

Trellix DLP (раніше McAfee)

DLP від Trellix (колишня McAfee) тісно інтегрується з екосистемою XDR, пропонуючи захист кінцевих точок, мережі та керування хмарним трафіком. Розроблений для великих підприємств, він забезпечує автоматизовані робочі процеси та розширене застосування політик, що відповідають сучасним стандартам відповідності, таким як GDPR та HIPAA. Trellix DLP відомий своєю масштабованістю та комплексною інтеграцією в центри операцій безпеки (SOC), що забезпечує цілісну видимість. Однак його інтерфейс користувача є складним, і організації повинні інвестувати в кваліфікованих адміністраторів для його розгортання та налаштування.

Переваги: Сильна інтеграція з XDR, масштабованість для підприємств, готовність до дотримання вимог.

Недоліки: Складний інтерфейс, ресурсомісткий, вимагає кваліфікованого персоналу.

Nightfall AI

Nightfall AI – це хмарна DLP-платформа, спеціально розроблена для середовищ, орієнтованих на SaaS. Вона інтегрується через API у Slack, GitHub, Google Drive та Microsoft 365, пропонуючи класифікацію конфіденційних даних, таких як PII, PHI та фінансова інформація, за допомогою штучного інтелекту в

режимі реального часу. Її перевага полягає в низькому рівні хибних спрацьовувань, швидкому розгортанні без агентів та комплексній підтримці хмарних систем відповідності, включаючи GDPR та SOC 2. Nightfall особливо приваблива для стартапів та середніх фірм, які застосовують хмарні стратегії, але їй бракує DLP на мережевому рівні та глибшого контролю кінцевих точок.

Переваги: Швидке розгортання на основі API, розширене виявлення за допомогою штучного інтелекту, дизайн, орієнтований на SaaS.

Недоліки: Обмежено SaaS/хмарою, немає традиційного мережевого DLP.

Data Loss Prevention Microsoft Purview

Microsoft Purview – це нативне хмарне рішення DLP для екосистеми Microsoft. Воно інтегрується з програмами Microsoft 365, Teams і SharePoint, забезпечуючи автоматизовану класифікацію даних, маркування та забезпечення відповідності вимогам. Для організацій, які вже інтегровані в стек Microsoft, Purview пропонує безперешкодне впровадження та шаблони для дотримання вимог GDPR, HIPAA та SOC 2. Однак воно надає обмежену функціональність поза середовищами Microsoft, що може бути обмежувальним для гібридних або багатохмарних компаній.

Переваги: безперебійна інтеграція з M365, автоматизована відповідність вимогам, зручні шаблони.

Недоліки: вузька зосередженість на екосистемі Microsoft, обмежена підтримка кількох хмар.

Google Cloud DLP

Google Cloud DLP зосереджений на класифікації та деідентифікації конфіденційних даних на основі штучного інтелекту/машинного навчання в GCP та Google Workspace. Він надає потужні хмарні API для розробників та підприємств для виявлення та захисту структурованих та неструктурованих даних. Ідеально підходить для організацій, які активно інвестують у Google Cloud, підтримує відповідність GDPR, HIPAA та SOC 2, але йому бракує ширшої підтримки кількох хмар.

Переваги: Сильна класифікація ШІ, легка інтеграція з GCP/Workspace.

Недоліки: Обмежений охоплення поза екосистемою Google.

Netskope DLP

Netskope інтегрує CASB, SWG та DLP в єдину платформу SASE, пропонуючи детальний контроль веб-трафіку та SaaS. Вона забезпечує вбудоване сканування, виявлення на основі штучного інтелекту та забезпечення дотримання політик, що робить її найкращим вибором для організацій, орієнтованих на хмарні технології. Netskope DLP має сильні сторони у сфері хмарної відповідності та прозорості SaaS, але відомий своєю високою вартістю та складністю розгортання.

Переваги: Відмінна SaaS/веб-прозорість, хмарно-орієнтований, відповідає вимогам.

Недоліки: Дорого, складно впроваджувати у великих масштабах.

Digital Guardian (Fortra)

Digital Guardian забезпечує надійний захист кінцевих точок та мережевих даних, зосереджуючись на відстеженні походження даних та захисті інтелектуальної власності. Він підтримує гібридне розгортання та регуляторні рамки, такі як GDPR, HIPAA, SOC 2, що робить його популярним серед виробничих та медичних підприємств. Його детальна звітність та криміналістика допомагають розслідувати внутрішні ризики, але система є важкою та вимагає виділених ресурсів.

Переваги: Надійне гібридне розгортання, глибока аналітика, охоплення відповідності вимогам.

Недоліки: Застарілий інтерфейс користувача, крута крива навчання, вимогливий до ресурсів.

DLP Palo Alto Networks

Palo Alto Networks інтегрує DLP у свою екосистему Prisma Access та NGFW, забезпечуючи вбудовану перевірку трафіку в хмарі, мережі та на кінцевих точках. Це допомагає підприємствам захищати дані під час руху,

застосовувати політики в SaaS та веб-трафіку, а також відповідати вимогам, таким як GDPR та HIPAA. Його перевага полягає в синергії з архітектурою Zero Trust Palo Alto, що робить його привабливим для організацій, які вже інвестували в брандмауери PANW та Prisma Cloud. Однак його вартість та складність роблять його менш придатним для малого та середнього бізнесу.

Переваги: безперешкодна інтеграція з PANW, видимість на межі хмари, готовність до дотримання вимог.

Недоліки: Дорого, вимагає особливих навичок для розгортання та обслуговування.

Code42 Incydr

Code42 Incydr спеціально розроблений для управління інсайдерськими ризиками, зосереджуючись на відстеженні витоку даних та ризикованого обміну файлами між кінцевими точками та хмарними додатками. Замість того, щоб сильно покладатися на перевірку контенту, він забезпечує аналіз походження даних та видимість активності користувачів. Це робить його особливо ефективним для виявлення навмисних інсайдерських витоків, тіньових IT-систем або недбалих співробітників, які порушують правила безпеки даних. Він інтегрується з інструментами SIEM та підтримує звітність про відповідність. Однак його обмежена класифікація на основі штучного інтелекту та відсутність розширеного контролю SaaS роблять його менш комплексним, ніж повноцінні інструменти DLP.

Переваги: Відмінне виявлення внутрішніх загроз, просте розгортання, чітка криміналістика.

Недоліки: Обмежена класифікація, слабші SaaS/хмарні функції.

Endpoint Protector (CoSoSys)

Endpoint Protector спеціалізується на DLP на рівні пристроїв, пропонуючи надійний контроль над USB-портами, зовнішніми сховищами та принтерами. Він працює у Windows, macOS та Linux, що робить його ідеальним для різноманітних IT-середовищ. Завдяки хмарному та локальному розгортанню він дозволяє

малому та середньому бізнесу (SMB) та підприємствам запобігати випадковим витокам даних, одночасно підтримуючи відповідність нормативним вимогам, таким як GDPR та HIPAA. Його сильною стороною є видимість кінцевих точок, але йому бракує розширеного хмарного DLP, і може знадобитися зусилля для точного налаштування політик для випадків використання на рівні підприємства.

Переваги: Надійний контроль пристроїв та кінцевих точок, кросплатформна підтримка.

Недоліки: Обмежені можливості SaaS/хмари, складність початкового налаштування.

Safetica

Safetica – це економічно ефективний інструмент для захисту від втрати даних (DLP) та управління внутрішніми ризиками для малого та середнього бізнесу. Він забезпечує моніторинг кінцевих точок, аналіз активності співробітників та класифікацію даних для запобігання витокам, спричиненим внутрішніми загрозами або недбалістю. Safetica відома швидким розгортанням та зручними панелями інструментів, що робить її доступною навіть для організацій без великої IT-команди. Вона також підтримує такі системи відповідності, як GDPR та HIPAA, хоча її масштабованість та розширені хмарні інтеграції обмежені порівняно з корпоративними постачальниками.

Переваги: Доступна ціна, проста у використанні, видимість інсайдерських ризиків.

Недоліки: Обмежена масштабованість підприємства, слабша інтеграція SaaS.

DLP GTB Technologies

GTB Technologies зосереджується на DLP на основі політик, використовуючи готові шаблони відповідності та глибоку інтеграцію з платформами SIEM. Вона забезпечує контроль кінцевих точок, мережі та хмари, що робить її придатною для регульованих галузей, таких як банківська справа та охорона здоров'я. GTB вирізняється перевіркою контенту в режимі реального

часу та гібридними варіантами розгортання. Однак, як менший постачальник, вона має меншу ринкову популярність та менше тематичних досліджень клієнтів порівняно зі світовими лідерами.

Переваги: Надійні шаблони відповідності, гнучке гібридне розгортання.

Недоліки: Обмежена впізнаваність бренду, менше рекомендацій від підприємств.

Data Loss Prevention Zscaler

Zscaler вбудовує DLP у свою платформу Zero Trust Exchange, що дозволяє здійснювати вбудоване сканування SaaS, веб- та електронного трафіку. Вона розроблена для хмарних розгортань, що робить її ідеальною для підприємств з розподіленими командами та середовищами з високим рівнем SaaS. Завдяки класифікації на основі штучного інтелекту Zscaler гарантує, що конфіденційні дані не будуть витікати через неавторизовані програми або канали. Однак, його зосередженість на хмарі означає меншу глибину традиційного DLP для кінцевих точок.

Переваги: хмарний, підхід Zero Trust, сильний SaaS/веб-контроль.

Недоліки: Обмежена видимість кінцевих точок, високі витрати при великому масштабі.

Data Loss Prevention Acronis DeviceLock

Acronis DeviceLock розроблений для керування кінцевими точками та USB-пристроями, забезпечуючи детальний моніторинг передачі даних, принтерів та офлайн-сценаріїв. Він легкий, ефективний та особливо корисний для організацій із суворими вимогами до контролю над знімними носіями. Хоча він допомагає досягти відповідності GDPR/HIPAA, йому бракує розширених хмарних інтеграцій DLP та SaaS, що може обмежувати впровадження в сучасних гібридних середовищах.

Переваги: Надійний захист на рівні пристрою, доступна ціна, легке розгортання.

Недоліки: Слабке покриття SaaS/хмари, обмежені розширені функції.

Data Loss Prevention Check Point

Check Point надає DLP-рішення як частину свого стеку безпеки електронної пошти та шлюзів. Він відстежує вихідну електронну пошту та контент, що поширюється через веб-трафік, зменшуючи випадкові витоки та ризики фішингу. Найкраще підходить для організацій, які прагнуть забезпечити дотримання вимог щодо електронної пошти (GDPR, HIPAA). Однак, він не забезпечує широкого охоплення кінцевих точок або SaaS, що робить його скоріше доповненням, ніж повним пакетом DLP.

Переваги: Надійний захист електронної пошти, проста інтеграція.

Недоліки: Орієнтовано лише на електронну пошту/шлюз, бракує видимості кінцевих точок.

Data Loss Prevention Cyberhaven

Cyberhaven переосмислює DLP за допомогою технології передачі даних, відстежуючи, як конфіденційна інформація переміщується між кінцевими точками, SaaS-додатками та хмарними сховищами. Замість простого блокування, він забезпечує глибоке розуміння ризикованих потоків, допомагаючи запобігати внутрішнім загрозам та тіньовим IT-атакам. Завдяки управлінню внутрішніми ризиками в режимі реального часу, Cyberhaven підтримує дотримання вимог, надаючи командам SOC розширену аналітику. Будучи новішим рішенням, його екосистема менша порівняно з існуючими операторами.

Переваги: Унікальний підхід до походження даних, видимість внутрішніх загроз.

Недоліки: новіший постачальник, менша екосистема, менш перевірена масштабованість.

OpenDLP (з відкритим вихідним кодом)

OpenDLP – це локальний інструмент DLP з відкритим кодом, розроблений для сканування баз даних і файлових систем для виявлення конфіденційних даних. Він забезпечує гнучкість для організацій з обмеженим бюджетом або тих, хто хоче налаштувати свій DLP. Хоча він підтримує базову звітність про

відповідність вимогам, йому бракує хмарних інтеграцій, класифікації штучного інтелекту та розширених функцій управління внутрішніми ризиками, що робить його менш практичним для сучасних підприємств.

Переваги: Безкоштовний, настроюваний, добре підходить для тестових середовищ.

Недоліки: Застарілий, бракує підтримки, немає можливостей SaaS/хмари.

MyDLP (з відкритим кодом)

MyDLP – це застарілий інструмент із відкритим кодом для запобігання втраті даних, який забезпечує базовий моніторинг електронної пошти, веб-трафіку та кінцевих точок. Він підходить для невеликих організацій, які експериментують із концепціями DLP або потребують спрощеної підтримки відповідності без високих витрат. Хоча він охоплює багатоканальний моніторинг, розробка сповільнилася, і йому бракує сучасних хмарних функцій, класифікації штучного інтелекту та аналітики внутрішніх ризиків. Тим не менш, він може слугувати рішенням початкового рівня для малого та середнього бізнесу з локальною інфраструктурою.

Переваги: Безкоштовне, просте багатоканальне висвітлення, корисне для тестування.

Недоліки: Застаріла, обмежена підтримка, відсутність SaaS або розширених інструментів для забезпечення відповідності.

Data Loss Prevention Trend Micro

Trend Micro інтегрує DLP у свої рішення для захисту кінцевих точок, шлюзи електронної пошти та гібридні хмарні рішення для безпеки. Це особливо ефективно для малих і середніх підприємств, які вже використовують антивірус або захист кінцевих точок Trend Micro, надаючи їм легке розширення DLP. Це допомагає організаціям дотримуватися вимог GDPR та HIPAA, але не має розширеного відстеження походження даних та сучасних інтеграцій SaaS. Trend Micro є надійним, але кращим як додатковий інструмент, ніж окремий корпоративний DLP.

Переваги: надійний постачальник, сильна інтеграція кінцевих точок/електронної пошти, легке впровадження.

Недоліки: Обмежені інновації хмарного DLP, слабше покриття інсайдерських ризиків.

Data Loss Prevention SecureTrust

SecureTrust DLP – це рішення, орієнтоване на відповідність вимогам, розроблене для малого та середнього бізнесу, якому потрібні доступні та ефективні засоби контролю безпеки даних. Воно підтримує моніторинг кінцевих точок, мережі та електронного трафіку за допомогою попередньо створених шаблонів відповідності GDPR, HIPAA та SOC 2. Його легкий дизайн спрощує розгортання для невеликих IT-команд. Однак йому бракує ширших корпоративних функцій, класифікації на основі штучного інтелекту та розширеного виявлення внутрішніх загроз порівняно з преміум-постачальниками.

Переваги: Доступні ціни, просте розгортання, шаблони, готові до дотримання вимог.

Недоліки: Обмежена масштабованість, слабші розширені функції.

Data Loss Prevention (DLP) Imperva

Imperva зосереджена на безпеці даних баз даних та веб-додатків, що робить її одним із найнадійніших варіантів для середовищ структурованих даних. Вона забезпечує моніторинг та забезпечення дотримання політик на рівні додатків, з потужною підтримкою відповідності для фінансових послуг та охорони здоров'я. Хоча Imperva інтегрується в гібридні розгортання, вона не є повноцінним хмарним DLP та має обмежену функціональність щодо кінцевих точок та внутрішніх ризиків. Найкраще підходить для підприємств, які надають пріоритет захисту баз даних над загальним моніторингом даних.

Переваги: Чудово підходить для баз даних, надійне покриття відповідності вимогам.

Недоліки: Дорого, бракує функцій кінцевих точок/SaaS, лише для підприємств.

Rapid7 InsightIDR (з DLP)

Rapid7 InsightIDR – це, перш за все, рішення для SIEM та XDR, але воно включає можливості DLP для моніторингу потоків конфіденційних даних. Його перевага полягає у співвіднесенні інцидентів втрати даних із ширшими сповіщеннями про безпеку, забезпечуючи цілісну видимість у SOC. Він підтримує гібридні середовища та звітність про відповідність, але не є спеціалізованою DLP-платформою. Організації часто обирають його, коли їм потрібні можливості SIEM та DLP в одному рішенні.

Переваги: інтеграція SIEM + DLP, потужна аналітика, зручність для SOC.

Недоліки: Не чистий DLP, дорогий, вимагає зрілого SOC.

Data Loss Prevention Qualys

Qualys розширює свою платформу управління вразливостями базовими функціями хмарного захисту від вразливостей (DLP). Вона зосереджена на виявленні конфіденційних даних на кінцевих точках та хмарних ресурсах, інтегруючись зі скануванням вразливостей для уніфікованої звітності. Хоча вона допомагає виконувати вимоги до відповідності, вона не є повнофункціональною DLP і не має розширеного управління внутрішніми ризиками або класифікації на основі штучного інтелекту. Вона найкраще підходить для організацій, які вже інвестували в екосистему Qualys.

Переваги: Інтеграція зі скануванням вразливостей, легке розгортання.

Недоліки: Обмежена цінність автономного захисту від втрати даних, слабші функції SaaS.

Splunk ES (доповнення DLP)

Splunk Enterprise Security можна розширити за допомогою плагінів DLP, що забезпечують моніторинг конфіденційних даних у робочих процесах SIEM. Він використовує аналіз походження даних та розширену аналітику для співвіднесення руху даних із ширшими загрозами. Це робить його потужним рішенням для підприємств зі зрілими SOC. Однак він вимагає багато ресурсів і не є окремим рішенням DLP. Організації повинні вже запускати Splunk у великих

масштабах, щоб скористатися перевагами цієї інтеграції.

Переваги: Розширена аналітика, синергія SIEM+DLP, сильний потік даних.

Недоліки: Складний, дорогий, вимагає інфраструктури Splunk.

Пакет програмного забезпечення для обробки даних Fortra

Корпоративний пакет Fortra надає комплексні функції захисту від втрати даних (DLP) для кінцевих точок, мережі та хмари. Він підтримує розширену класифікацію штучного інтелекту, виявлення внутрішніх ризиків та рамки відповідності, такі як GDPR, HIPAA та SOC 2. Його перевага полягає у великомасштабному розгортанні в гібридних середовищах. Однак його складність та вимоги до ресурсів роблять його більш придатним для великих підприємств, ніж для малого та середнього бізнесу.

Переваги: Повне покриття корпоративного рівня, гібридне розгортання, готовність до дотримання вимог.

Недоліки: Складний, ресурсомісткий, дорогий.

DLP управління даними Dow Jones

Dow Jones пропонує нішеве рішення для захисту даних (DLP), орієнтоване на управління даними, яке забезпечує надійну аналітику відповідності та моніторинг для структурованих даних. Воно використовується в галузях, де нормативні вимоги вимагають детального нагляду. Хоча воно додає цінності в контексті, що вимагає складного управління, воно не є універсальним хмарним DLP та не має широкої інтеграції з кінцевими точками та SaaS.

Переваги: Потужні інструменти управління, аналітика, орієнтована на дотримання вимог.

Недоліки: Вузький сценарій використання, обмежена універсальність, менша база впровадження.

Seclore Data-Centric Security

Seclore поєднує управління цифровими правами (DRM) із DLP, захищаючи файли на рівні даних за допомогою шифрування, контролю доступу

та забезпечення дотримання політик. Він підтримує гібридне розгортання та допомагає організаціям досягати відповідності, гарантуючи захист конфіденційних документів навіть за межами мереж компанії. Хоча його орієнтований на дані підхід є потужним, складне ліцензування та інтеграція можуть уповільнити впровадження.

Переваги: Безпека на рівні файлів, надійне поєднання DRM+DLP.

Недоліки: Складне ліцензування, складна інтеграція, незручний для малого та середнього бізнесу.

Рекомендації:

– Підприємства → Symantec, Forcepoint, Proofpoint.

– Користувачі хмарного SaaS → Nightfall AI, Netskope, Microsoft Perview.

– Малий та середній бізнес → Kickidler, Safetica, Endpoint Protector.

– Організації, що орієнтовані на дотримання вимог → Digital Guardian, GTB, Imperva.

– Управління інсайдерськими ризиками → Code42, Kickidler, Cyberhaven.

На відміну від застарілих постачальників DLP для корпоративних користувачів, Kickidler поєднує моніторинг кінцевих точок з аналітикою внутрішніх ризиків, віддаленим контролем співробітників та аналізом продуктивності працівників. Це робить його унікальним гібридом між DLP та аналітикою робочої сили, що особливо цінно для малого та середнього бізнесу та розподілених команд.

– Моніторинг внутрішніх загроз – запис екрана, реєстрація натискань клавіш.

– Доступна ціна для малого та середнього бізнесу – нижча вартість порівняно з корпоративними гігантами.

– Легке розгортання для віддалених команд – швидке налаштування, хмара + локальна платформа.

– Аналітика продуктивності співробітників – моніторинг часу, завдань та ефективності.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

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

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

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

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

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

Переваги та недоліки Python

Переваги:

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

– Підвищує продуктивність. Це дуже продуктивна мова. Завдяки її простоті розробники можуть зосередитися на розв’язанні проблеми. Їм не

потрібно витрачати багато часу на розуміння синтаксису або поведінку мови програмування. Ви пишете менше коду та виконуєте більше завдань.

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

– Динамічно типізована. Python не визначає тип змінної, доки ми не запустимо код. Вона автоматично надає тип даних, коли відбувається процес виконання. Фахівець може не турбуватися про оголошення змінних та типи даних.

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

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

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

Недоліки:

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

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

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

– Доступ до бази даних. Програмувати на цій мові легко, але коли ми взаємодіємо з базою даних, її не вистачає. Рівень доступу до бази даних у Python примітивний та недостатньо розвинений у порівнянні з іншими популярними технологіями.

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

Застосування Python:

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

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

– Для машинного навчання. Машинне навчання (ML) є основою більшості завдань науки даних. Він є областю штучного інтелекту, пов’язаною з

використанням алгоритмів, що дозволяють машинам вивчати закономірності та тенденції на основі історичних даних, щоб робити прогнози на основі невідомих даних. – Використовуючи методи ML, ми можемо створювати моделі, які можуть точно передбачити швидкість відтоку клієнтів компанії, оцінити ризик виникнення у людини певного захворювання, визначити оптимальне розташування автомобілів таксі й т.д. За допомогою Python ми можемо побудувати модель ML, використовуючи лише три рядки коду.

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

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

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

Яким фахівцям потрібно володіти Python:

– Фахівець з даних.

– Аналітик даних.

– Інженер даних.

– Інженер з машинного навчання.

– Журналіст даних.

– Архітектор даних.

– Повний стек веб-розробника.

– Backend-розробник.

– DevOps-інженер.

– Інженер-програміст.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи реалізації інтелектуального DLP-агента.

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) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

DLP став обов'язковим рішенням для компаній будь-якого розміру. Правильний вибір залежить від розміру компанії, інфраструктури та потреб у дотриманні вимог.

Коли люди кажуть «швидке забезпечення безпеки», вони часто мають на увазі швидке введення. Це має значення, але це лише половина справи. У більшості українських організацій, найпоширеніший збій є нудним і дорогим: витік даних.

Ось як це виглядає на практиці:

– Агент служби підтримки вставляє повний ланцюжок запитів у помічник GenAI, включаючи адреси та дані облікового запису.

– Розробник надає доступ до трасування стека, яке містить токени або ключі API.

– Торговий представник додає список клієнтів у запит, щоб «підсумувати персони».

Жоден з них не є шкідливим. Це звичайні робочі звички, які тепер передаються через інструменти, що ніколи не були розроблені як безпечні склепіння даних.

Ідея інтелектуального DLP-агента проста: ставитися до запитів GenAI як до будь-якого іншого каналу вихідних даних. Це означає ідентифікацію типів конфіденційного контенту (PII, PHI, PCI, облікові дані, секрети), а потім застосування дій політики, перш ніж дані залишать будівлю.

Якщо ваша програма не може відповісти на запитання «хто що вставив, куди і коли», у вас немає управління. У вас є надія.

Як інтелектуальний DLP-агент працює для безпеки запитань LLM

Спрощений вигляд перевірки та редагування перед надсиланням, створений за допомогою штучного інтелекту.

Інтелектуального DLP-агента найкраще розуміти як DLP на основі політик у SaaS, кінцевих точках, браузерах та інструментах штучного інтелекту. Станом на початок 2026 року, публічні нотатки про продукт підкреслюють охоплення великого набору SaaS-додатків (понад 100), а також елементів керування кінцевими точками та браузером. Ця широта охоплення важлива, оскільки оперативне використання рідко залишається в межах одного санкціонованого додатка.

Коли я оцінюю інтелектуального DLP-агента на предмет оперативної безпеки, я зосереджуюся на трьох механіках:

1. Якість виявлення реального контенту підказки. Підказки хаотичні. Вони змішують контекст, журнали, фрагменти коду та вставлені таблиці. Цінність інтелектуального DLP-агента залежить від виявлення конфіденційних даних у цьому хаосі, водночас зберігаючи достатньо низький рівень шуму, щоб люди не обходили його.

У проекті я протестував детектори на:

– Поширені шаблони ідентифікаційних даних України (імена та адреси, шаблони, подібні до номерів соціального страхування, ідентифікатори співробітників).

– Дані про оплату (номери карток на скріншотах або вклеєних квитанціях).

– Секрети (ключі API, токени, закриті ключі).

– Витоки «випадкового джерела» (файли конфігурації, вміст.env, журнали налагодження).

2. Політичні дії, що відповідають принципам роботи команд. Захист від втрати інформації (DLP) не корисний, якщо він лише кричить. Вам потрібні дії, що відповідають ризику та контексту. інтелектуального DLP-агента зазвичай

передбачає такі дії, як блокування, видалення, карантин, видалення або попередження та інструктаж.

Для підказок LLM мені потрібно мати принаймні два безпечних значення за замовчуванням:

– Викресліть та врахуйте середній ризик (наприклад, видаліть токен, але залиште решту підказки корисною).

– Блокування та перенаправлення на перевірку щодо високого ризику (наприклад, захищена медична інформація, повні дані про платежі або експорт великих клієнтів).

3. Видимість тіньового ШІ та розповсюдження GenAI. Більшість організацій не мають одного застосунку GenAI. У них їх двадцять. Оновлення інтелектуального DLP-агента 2026 року передбачають такі функції, як аналіз використання застосунків та робочі процеси розслідування в стилі криміналістики для реконструкції інсайдерських ризиків. Я вважаю їх «приємним мати», поки це не доведено, але напрямок правильний: вам потрібна карта потоку підказок.

Якщо ви створюєте власні потоки (RAG-конвеєри, підсумовувачі заявок, внутрішні копілоти), підхід інтелектуального DLP-агента на стороні розробника пояснюється на їхній сторінці інтелектуального DLP-агента для розробників. Для мене ключове питання полягає в тому, чи можна застосовувати ті самі політики в користувацьких сервісах, що й у SaaS.

Що мені подобається в підході інтелектуального DLP-агента, так це те, що він відповідає тому, як проявляється ризик. У швидких витоках проблемою є корисне навантаження, а не лише поведінка моделі. DLP – це правильний клас інструментів для зупинки «скопійованого витоку».

Тим не менш, я врахував ці обмеження:

– Хибнопозитивні результати – це операційний борг: навіть хороший детектор виявить граничні випадки (тестові картки, підроблені облікові дані, дані навчання безпеки). Вам знадобиться налаштування та обробка винятків.

– Досвід користувача має значення: якщо блокування здається випадковим, люди переключатимуться на особисті пристрої або несанкціоновані інструменти. Попередження у стилі коучингу часто працюють краще, ніж жорсткі блокування на початку.

– Ціноутворення не є самообслуговуванням: станом на лютий 2026 року ціноутворення інтелектуального DLP-агента зазвичай базується на котируваннях і залежить від користувачів, програм та обсягу даних. Це нормально для корпоративного DLP, але це уповільнює складання бюджету.

Інтелектуальний DLP-агент – це не єдиний рівень, який вам потрібен. Це рівень, який є надійним для певного класу ризику: конфіденційні дані, що передаються через сучасні канали співпраці та штучного інтелекту.

Якщо ваша модель загроз включає джейлбрейки та атаки із застосуванням інструментів, поєднайте DLP з рівнем середовища виконання LLM.

План розгортання

Я не починав з «заблокувати все». Я почав з вимірювання, а потім зробив:

– Відкриття: Визначення того, які програми штучного інтелекту використовуються, ким і звідки (браузер, кінцева точка, корпоративна мережа).

– Монітор: Запустіть у режимі лише сповіщень протягом короткого періоду, а потім виміряйте, що насправді позначається.

– Налаштування: Зменшення шуму за допомогою політик з обмеженою областю дії (для кожного відділу, для кожної програми, для кожного типу даних).

– Забезпечити дотримання: Спочатку блокуйте лише категорії з найвищим рівнем ризику, а потім поступово розширюйте охоплення.

– Доведення: Створіть прості показники, які розуміють керівники (блокування витоків за типом, економія часу на розслідування, зменшення кількості ручних перевірок).

Така послідовність зберігає довіру. Щойно користувачі довіряють послідовності, забезпечення дотримання стає легшим.

3.2 Розробка структурної схеми

Протягом понад десятиліття система запобігання втратам даних (DLP) позиціонувалася як остання лінія захисту від витоків даних. Однак напередодні 2026 року більшість лідерів у сфері безпеки тихо визнають правду: традиційна DLP більше не відповідає тому, як дані створюються, доступні та розкриваються.

Хмарні архітектури, розростання SaaS, доступ на основі штучного інтелекту та зростання неструктурованих даних докорінно змінили проблему. Результат? Збільшення розриву між традиційним DLP та сучасним DLP, а також зростаючий зсув до DSPM, масштабного виявлення даних та контексту на основі штучного інтелекту як справжніх альтернатив DLP.

Застаріле DLP не сприймає сучасний ландшафт даних. DSPM + виявлення даних + автоматизація забезпечують прозорість та контроль, які дійсно потрібні лідерам у сфері безпеки.

Для чого був створений традиційний DLP

Традиційна система захисту від втрати даних (DLP) виникла в зовсім іншу епоху безпеки. Її основні припущення були простими:

– Дані зберігаються у відомих місцях (кінцеві точки, електронна пошта, мережеві шлюзи).

– Конфіденційні дані можна ідентифікувати за допомогою політик на основі сигнатур.

– Блокування або сповіщення про переміщення даних запобігає втраті.

По суті, традиційний DLP спирається на:

– Детерміністичне виявлення (точні збіги, регулярні вирази, відбитки пальців).

– Статичні правила політики.

– Вбудовані засоби контролю запобігання (блокування, карантин, шифрування).

Це працювало досить добре, коли потоки даних були передбачуваними, а

інфраструктура централізованою. Але ці припущення вже не відповідають дійсності.

Чому застарілі системи захисту від втрати даних неефективні у 2026 році

1. Дані вийшли за межі периметра

Сучасні організації працюють за такими напрямками:

– SaaS-платформи.

– Хмарні сховища даних.

– Інструменти для співпраці.

– Штучний інтелект, що працює як копілоти, та інтеграції зі сторонніми розробниками.

Запобігання втраті даних у хмарі більше не полягає в зупинці файлів на шлюзі. Доступ до даних здійснюється безпосередньо через API, ідентифікатори та програми – часто без будь-якого «переміщення».

Традиційна DLP-технологія не може захистити те, що вона не бачить.

2. Політики на основі підписів не масштабуються

Застарілі системи захисту від втрати даних значною мірою залежать від політик на основі сигнатур:

– Регулярний вираз для ідентифікаційної інформації.

– Статичні класифікатори.

– Попередньо визначені шаблони вмісту.

Ці підходи не дають результату, коли:

– Дані напівструктуровані або неструктуровані.

– Контекст визначає чутливість.

– Одні й ті ж дані є конфіденційними в одному випадку використання та нешкідливими в іншому.

Це основне обмеження детерміністичного та контекстно-залежного виявлення. Детерміновані правила є крихкими; сучасні середовища – ні.

3. Втома від тривоги замінює реальне зниження ризику

Більшість програм DLP виходять з ладу не тому, що вони пропускають сповіщення, а тому, що їх генерують забагато.

Команди безпеки стикаються з:

– Тисячі сповіщень про низьку достовірність.

– Мало ясності щодо радіуса вибуху або експозиції.

– Ручне сортування без пріоритетів.

Без контекстно-залежної оцінки ризиків, DLP стає радше шумом, ніж захистом.

4. DLP ніколи не створювався для управління неструктурованими даними

Сьогодні більшість конфіденційних даних зберігається в:

– Документи.

– Електронні таблиці.

– Повідомлення чату.

– Вікі.

– Репозиторії коду.

Традиційне DLP не було розроблено для масштабного управління неструктурованими даними. Йому бракує:

– Усвідомлення власності.

– Бізнес-контекст.

– Видимість життєвого циклу.

Блокування даних не вирішує проблему розростання даних.

Як виглядатиме сучасний захист даних у 2026 році

Сучасний захист даних повністю змінює модель DLP.

Замість того, щоб питати:

«Чи можемо ми заблокувати цю дію?»

Керівники служб безпеки зараз запитують:

«Чому ці дані розголошуються, хто може отримати до них доступ і який реальний ризик?»

Основні принципи сучасного DLP

Сучасне DLP – це не окремий інструмент, а стратегія, побудована на видимості, контексті та автоматизації.

Він включає:

– Масштабне виявлення даних на хмарних, SaaS та штучно інтелектуальних платформах.

– Безпека на основі ідентифікації, яка пов'язує доступ з реальними користувачами та ролями.

– Безперервна оцінка ризиків, а не блокування в певний момент часу.

– Автоматизація політик замість керування статичними правилами.

Саме тут DSPM стає основоположним.

Від традиційного DLP до сучасної безпеки даних: чому DSPM є основоположним

Традиційний DLP: Контроль виходу

DSPM: Зменшення експозиції

DSPM (Управління безпекою даних) зосереджується на:

– Де знаходяться конфіденційні дані.

– Хто має до нього доступ.

– Чи є доступ надмірним, ризикованим чи непотрібним.

– Як змінюється експозиція з часом.

DSPM заповнює критичні прогалини, які пропускають традиційні DLP, забезпечуючи видимість у режимі реального часу, пріоритезацію ризиків та автоматизовані дії.

DSPM вирішує першопричину порушень: надмірне викриття даних, а не лише даних, що передаються.

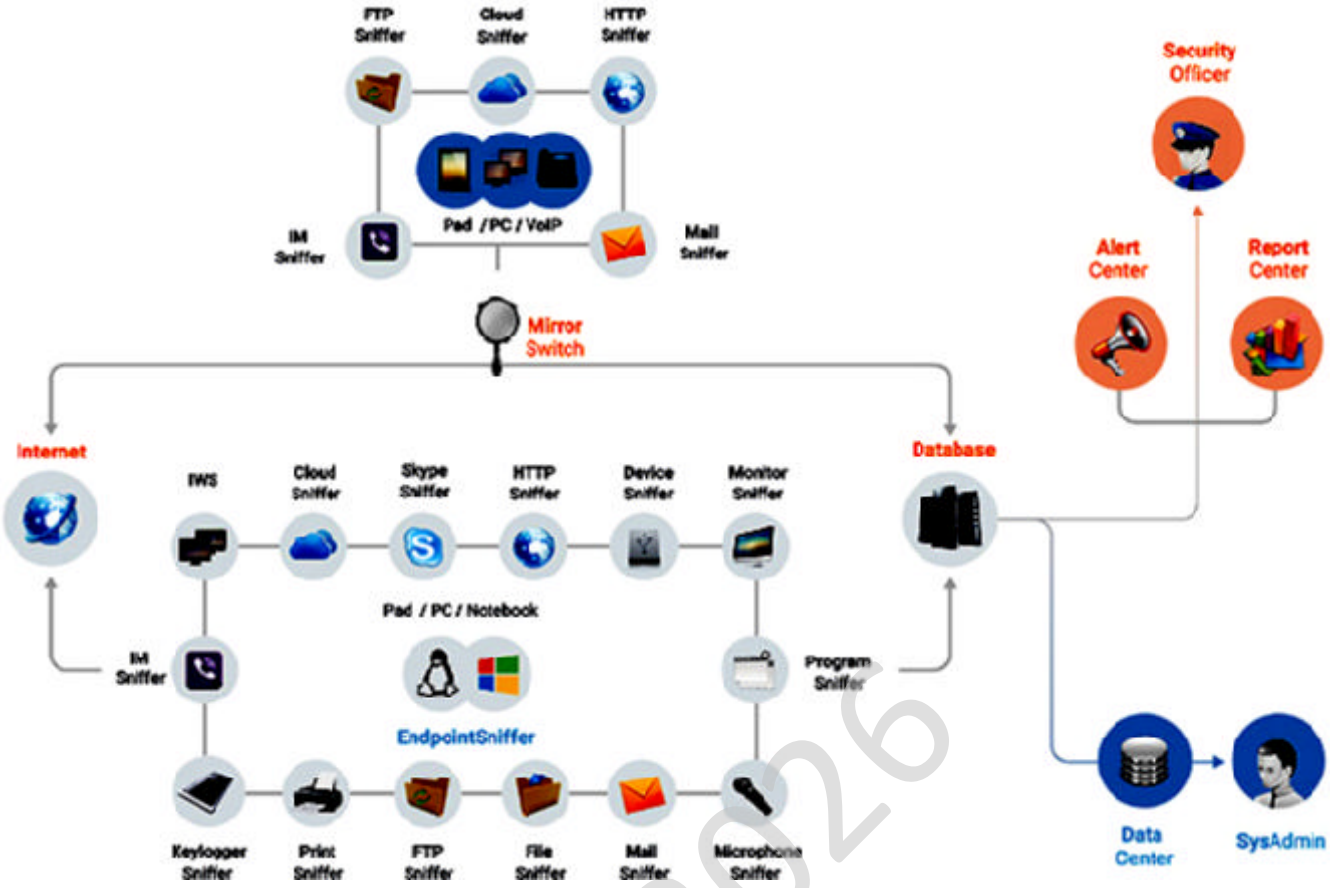

Рисунок 3.1 – Структурна схема системи

Технології, що замінять DLP у 2026 році

Сучасні організації у 2026 році впроваджують альтернативи DLP, які відповідають тому, як насправді працюють дані.

1. DSPM + Масштабне виявлення даних

Сучасні інструменти постійно виявляють:

– Структуровані та неструктуровані дані.

– Тіньові сховища даних.

– Забуті та осиротілі дані.

Платформа виявлення Інтелектуальний DLP-агент виявляє конфіденційні, регульовані та токсичні дані в усіх середовищах, щоб забезпечити дотримання принципів нульової довіри до даних.

2. Безпека з урахуванням ідентифікації

Замість блокування файлів, сучасний захист даних оцінює:

– Хто має доступ до даних.

– Звідки.

– Під якою роллю чи привілеєм.

– Чи є доступ виправданим.

Інтелектуальний DLP-агент забезпечує це для розповсюдження SaaS, підрядників, ідентифікаторів машин та агентного ШІ, надаючи інформацію в режимі реального часу про те, хто має доступ і чи це виправдано.

3. Виявлення на основі штучного інтелекту та з урахуванням контексту

Сучасні системи виходять за рамки детерміністичного виявлення та йдуть у бік:

– Поведінковий аналіз.

– Бізнес-контекст.

– Чутливість у використанні, а не лише у змісті.

Інтелектуальний DLP-агент виходить за рамки класифікації – використовує бізнес-контекст та поведінкову аналітику для виявлення ризиків високого впливу, а не лише аномалій.

4. Автоматизовані робочі процеси виправлення

Замість нескінченних сповіщень, сучасні платформи запускають:

– Зміна розміру дозволів.

– Видалення публічного посилання.

– Перепризначення права власності.

– Робочі процеси виправлення на основі політик.

Інтелектуальний DLP-агент автоматизує критично важливі робочі процеси, такі як визначення розміру прав доступу та перепризначення власності, завдяки чому команди безпеки можуть діяти масштабно, точно та швидко.

Роль штучного інтелекту в сучасній DLP-системі

Штучний інтелект не просто класифікує дані, а змінює спосіб доступу до них.

Зі копілотами, агентами та автономними робочими процесами:

– Дані запитуються, узагальнюються та повторно використовуються динамічно.

– Доступ є непрямим та безперервним.

– Традиційні елементи керування DLP повністю обходяться.

Сучасні платформи захисту даних призначені для:

– Шаблони доступу з урахуванням штучного інтелекту.

– Безперервна оцінка дозволів.

– Оцінка ризиків для агентного доступу до даних за допомогою штучного інтелекту.

Інтелектуальний DLP-агент створено для забезпечення цієї нової парадигми – оцінки динамічних шаблонів доступу та зменшення ризиків, пов'язаних зі штучним інтелектом, безперервно, а не лише реактивно.

Застаріле DLP просто не може працювати на цьому рівні.

Переосмислення «профілактики» у запобіганні втратам даних

Найбільша зміна мислення в традиційному DLP порівняно з сучасним DLP полягає в наступному:

Профілактика більше не означає блокування подій.

Профілактика означає усунення непотрібного впливу.

Від:

– Виявлення конфіденційних даних всюди.

– Проактивне обмеження доступу.

– Застосування принципів нульової довіри до даних.

– Автоматизація робочих процесів відновлення.

Сучасні організації запобігають порушенням ще до того, як вони взагалі відбудуться.

Сучасний захист даних також підтримує дотримання вимог до таких структур, як NIS2, DORA та Закон ЄС про штучний інтелект, забезпечуючи постійну прозорість, можливість аудиту та контроль ризиків.

Заключний висновок: DLP не помер, але його замінила стратегія

Традиційна система захисту від втрати даних (DLP) зазнала невдачі не тому, що служби безпеки неправильно її використовували.

Це не вдалося, оскільки ландшафт даних змінювався швидше, ніж модель.

У 2026 році сучасна система запобігання появі вірусу (DLP) більше не є окремим інструментом профілактики. Вона є результатом:

– DSPM.

– Виявлення даних у великих масштабах.

– Безпека з урахуванням ідентифікації.

– Контекстно-залежний інтелект на основі штучного інтелекту.

– Автоматизоване виправлення.

Для CISO, CIO та керівників із захисту даних питання більше не полягає в «Як налаштувати DLP?»

Це:

«Як нам захистити дані скрізь, де вони знаходяться, – перш ніж вони будуть втрачені?»

Це майбутнє сучасного захисту даних.

3.3 Розробка функціональної схеми

На функціональній схемі, що наведена на рисунку 3.2, показана реалізація функціональних можливостей системи реалізації DLP-агенту. Наведено результати виконання окремих етапів. Інтерес представляє ієрархічна організація баз шаблонів. Порівняння відбувається за принципом поступового наближення зі знаходження статистичної залежності чорно-білих зображень на основі кореляційного методу.

При цьому розмірність еталона й досліджуваного подання по вертикалі збігаються. Рух еталона уздовж вейвлет-образу сигналу відбувається в горизонтальному напрямку у випадку стаціонарного аналізу. При динамічному виявленні особливостей еталон і сигнал зіставляються правими границями.

Поява аномалій у різних точках спостереження протягом невеликого проміжку часу говорить про можливий взаємозв'язок подій. Об'єкт, що має більше ранній за часом момент виникнення аномалії може бути джерелом нестандартного поводження ряду даних. Досліджуючи виникнення аномалій на різних пристроях, можливим є побудова дерева аномалій, що веде від джерела через посередників різних рівнів до приймачів і навпаки: від приймача до джерела. Для реалізації функціональної схеми опишемо структуру методу динамічного виявлення аномалій в інтенсивностях потоків телекомунікаційних даних для статистичного модулю аналізу системи реалізації DLP-агенту, що включає наступні основні етапи:

1. Збір даних.

2. «Ручна» класифікація частини даних на предмет присутності нормального (типового) або аномального трафіку.

3. Попередній Фур'є-аналіз типового трафіку.

4. Виконання вейвлет-розкладання типового трафіку за обраними функціями для оцінки його спектрального-часового подання.

5. Обробка вейвлет-подання аномального трафіку.

а) визначення моментів різкої зміни характеру поводження рядів телекомунікаційних даних по рівневим і частотних характеристиках;

б) формування баз шаблонів вейвлет-образів:

– бази вейвлет-образів для характеру розвитку трафіку на тлі критичних ситуацій;

– бази вейвлет-образів для характеру розвитку трафіку до критичних моментів (до початку зареєстрованих проблем у роботі мережі);

– збереження відповідностей між даними баз;

в) визначення частотних показників для провокуючих розвиток аномалії подій, у тому числі аналіз енергетичних спектрів послідовностей до критичних моментів (спектра потужності й скалограми) на предмет перерозподілу енергії між частотами;

6. Верифікація прогнозу характеристик трафіку з реальними значеннями (на основі попередньо зібраних даних поза реальним режимом часу).

7. При надходженні нових даних:

а) зіставлення їх вейвлет-подань із шаблонами сформованих баз даних за допомогою кореляції й формування рішення про навність передкритичної або критичної ситуації.

б) виявлення зв'язків між аномаліями, що відбулися в різних точках спостереження.

У результаті виконання етапів з першого по шостий включно маємо:

– алгоритм розпізнавання позаштатної ситуації;

– початкові бази вейвлет-шаблонів для позаштатних ситуацій і попередніх за часом інтервалів;

– модель прогнозування характеру розвитку трафіку і як наслідок апріорне визначення можливих позаштатних ситуацій і їхня класифікація.

Застосування спектральних і спектрально-часових алгоритмів, є основними інструментами в розв'язуваному завданні динамічного виявлення аномалій у часових рядах інтенсивностей телекомунікаційних даних:

– аналіз внесків частот визначає формування моделей типового трафіку й окремих ділянок з особливостями;

– картини вейвлет-коефіцієнтів аномалій і попередніх їм ділянок сигналу формують бази даних шаблонів для характеру розвитку трафіку на тлі критичних ситуацій і до критичних моментів;

– побудовані на основі обчислених коефіцієнтів енергетичні спектри й інші залежності дозволяють відслідковувати розподіл (або перерозподіл) енергії між частотами й визначати сховану природу окремих ділянок сигналу.

Розкрито етапи нагромадження інформації, аналізу готових наборів, формування баз даних шаблонів, верифікації, виявлення аномалій у потоках даних, аналізу взаємозв'язків аномалій у різних потоках.

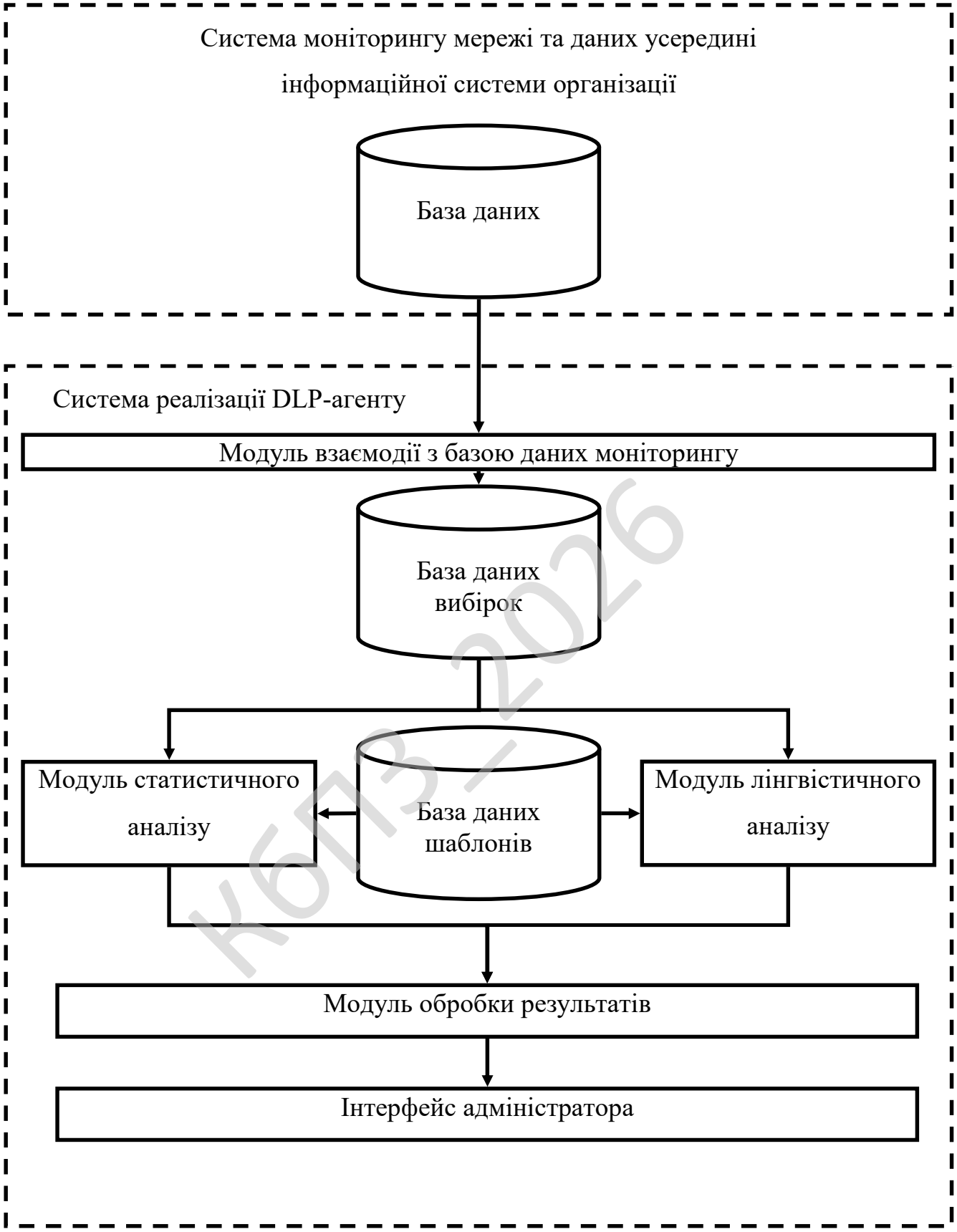

Рисунок 3.2 – Функціональна схема системи

Детально проаналізовані алгоритми дискретного перетворення Фур'є (ДПФ), дискретного перетворення Хартлі (ДПХ), швидкого перетворення Фур'є (ШПФ) і швидкого перетворення Хартлі (ШПХ).

Обґрунтовано можливість заміни широко використовуваного ДПФ на менш відоме, але більше ефективне ДПХ.

У результаті отримані математичні моделі оптимізації короткочасного аналізу (за участю часового «вікна» сканування) для прямих ПФ і ПХ зсувної послідовності.

Отримані практичні результати підтвердили ефективність запропонованих методів перетворення й у своєму загальному виді збіглися з теоретичними, незважаючи на присутні відхилення, внесені специфікою програмної реалізації. Відповідно до отриманих значень середніх часів виконання короткочасного спектрального аналізу в межах одного «вікна» можна зробити наступний вивід: для скорочення часу виконання короткочасного Фур'є-аналізу треба перше «вікно» обчислювати за допомогою ШПХ із наступним перетворенням обчислених значень у результат ПФ, а наступні «вікна» на основі запропонованого методу для ПФ.

Переваги запропонованого методу:

– висока швидкість обчислень;

– лінійна залежність швидкості обчислень від розміру «вікна».

Недоліки запропонованого методу:

– наявність швидко зростаючої погрішності обчислень ПХ після другої ітерації алгоритму перерахунку спектра.

Для усунення недоліку запропонованого методу необхідно:

– застосування розширених форматів подання даних;

– при досягненні критичного значення погрішності зробити перерахування спектра по стандартному алгоритмі.

Далі розглянемо спектрально-часові вейвлет-перетворенням (ВП). Розглянемо принципи різних варіацій дискретизованого перетворення, вплив

базису на результат вейвлет-розкладання й дані рекомендації з можливого використання вейвлетів.

Докладно розглянуті:

1) дискретне перетворення зі зміною масштабів і зрушень за значенням, рівному часової локалізації вейвлета;

2) дискретний діадний кратномасштабний аналіз;

3) дискретизоване розкладання.

Виконано аналіз виникаючих крайових ефектів спектрально-часових картин ВП на прикладах розгляду кутів впливу елементів аналізованої послідовності на вейвлет-коефіцієнти й кутів вірогідності (правдоподібності) у сітках перетворень. Усередині кута правдоподібності, як відомо, крайові ефекти проявлятися не можуть, оскільки кожна особливість, у тому числі й виникаюча на крах аналізованого сигналу, робить лише локальний вплив на вейвлет-спектрограмму. Скоректовано вид кутів впливу й правдоподібності для зсувної послідовності.

З позиції інформативності по кількості уточнюючих коефіцієнтів кращим є випадок 3), потім 2), потім 1).

З погляду збільшення області достовірних значень кращим є випадок 1), потім 3), потім 2).

У порядку зростання обчислювальної складності перетворення розташовуються в порядку 1), 2), 3).

Для завдання розпізнавання аномалії перші два критерії є вирішальними, причому по жодному з них обране перетворення не повинне бути гіршим.

Таким є дискретизоване перетворення, хоча його надмірність очевидна й підтверджується третім критерієм – обчислювальною складністю.

Можна припустити, що оптимальним було б перетворення, «яке знаходиться» між дискретним діадним і дискретизованим, причому саме перше повинне бути взяте за основу, так як має добре оптимізовані по організації обчислень і використовуваної пам'яті швидкі алгоритми статичного обчислення

сітки коефіцієнтів, і розширено для наділення перевагами дискретизованого розкладання. Результат може бути аналогічний використанню фреймів, так як перетворення на основі фреймів має велика подібність із дискретизованим розкладанням з тією різницею, що зміна масштабу відбувається по ступенях «двійки».

Розроблено схему виконання дискретизованого за часом і діадного по масштабах розкладання на основі швидкого алгоритму діадного кратномасштабного аналізу. Відзначимо, що в цей час існують алгоритми, оптимізучі стандартну схему швидкого діадного розкладання, наприклад на основі нестандартного матричного множення, тобто пропонований у роботі метод розрахунку вейвлет-коефіцієнтів може бути додатково оптимізований по обчислювальних витратах.

Виконано оптимізацію обчислень сіток дискретизованого розкладання за принципом фреймів і покладених у його основу алгоритмів діадного кратномасштабного аналізу при зрушенні послідовності.

Розроблено підхід до формування довірчої області на картині вейвлет-коефіцієнтів. Завдання було вирішено послідовним виконанням наступних етапів:

– вибір алгоритму пророкування значень майбутніх елементів часового ряду;

– розрахунок довірчого інтервалу для передвіщених значень;

– виконання верифікації;

– установлення зв'язку довірчого інтервалу із границею кута правдоподібності;

– розширення кута правдоподібності за допомогою введення довірчої області в картину вейвлет-коефіцієнтів.

Запропоновано підхід для мінімізації розкиду значень вейвлет-коефіцієнтів у розрахованій довірчій вейвлет-області на основі введення нестандартної внутрішньої границі довірчої часової області.

На впевненість влучення вейвлет-коефіцієнтів у розрахований діапазон (вейвлет-упевненість) впливають значення відрахунків вейвлета, так як їх можна інтерпретувати в рамках справжнього етапу як вагові коефіцієнти, відповідно до яких приймаються в розгляд відрахунки часового ряду, у тому числі неіснуючі.

Для використання в аналізі довірчої вейвлет-області найбільш підходящим є формування декількох її подань, заповнених, наприклад, значеннями:

– верхньої границі;

– нижньої границі;

– розрахованими на основі передвіщених значень апроксимованої часової послідовності;

– випадково розподіленими між верхньою й нижньою границями.

Комбінація таких подань довірчої вейвлет-області дозволить найбільше вірно судити про поводження вейвлет-коефіцієнтів за межами кута правдоподібності.

Для одержання картини вейвлет-коефіцієнтів, придатних для виконання операцій розпізнавання, необхідна певна довжина вихідної послідовності. Мала довжина послідовності спричиняється пропорційну кількість вейвлет-коефіцієнтів, що робить картину розкладання розбитої на відповідну кількість прямокутних областей. Елементи такої картини занадто грубі й непридатні для подальшого розпізнавання.

До вихідної послідовності пропонується застосувати алгоритм, що збільшує число елементів ряду. Проаналізовано результати використання спектральних і часових алгоритмів інтерполяції.

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

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу

сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей.

Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників.

Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

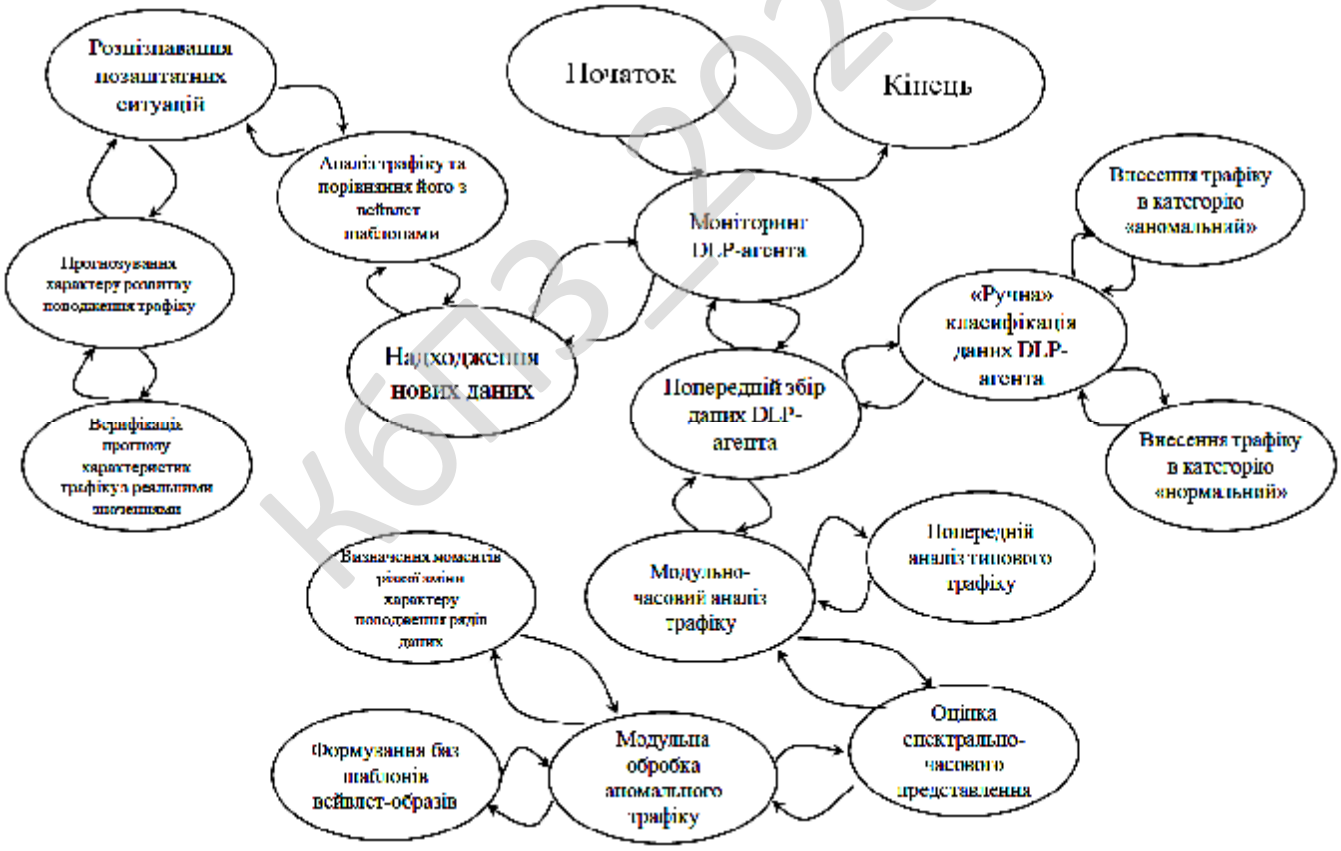

Рисунок 3.3 – Діаграма взаємодії процесів

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

При детальному її розгляді можна побачити як саме проходе взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

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

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз’яснення який саме тип блок-схем використовується. Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо.

Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними. У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач. Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції. Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елементу.

Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елементу. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи. Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані. Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи). З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення. Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і

детальної блок-схеми залежить результат всієї програми. При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об’єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких 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 підсистеми 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.

Система керування базами даних (СКБД, Database Management System, DBMS) – набір взаємопов'язаних даних (база даних) і програм для доступу до цих даних. Надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

Першим поколінням СКБД прийнято вважати ієрархічні й мережеві системи. Ці системи отримали широке поширення в 1970-х роках, а першою комерційною системою цього типу була система IMS компанії IBM.

У 1980-х роках ці системи були витіснені системами другого покоління – повсюдно використовуваними і донині реляційними СКБД. У цих системах використовувалися непроцедурні мови управління даними (SQL) і передбачався значний ступінь незалежності даних. Реляційні системи внесли значні удосконалення в управління даними: графічний користувацький інтерфейс (GUI), клієнт-серверні застосунки, розподілені бази даних, паралельний пошук даних та інтелектуальний аналіз даних.

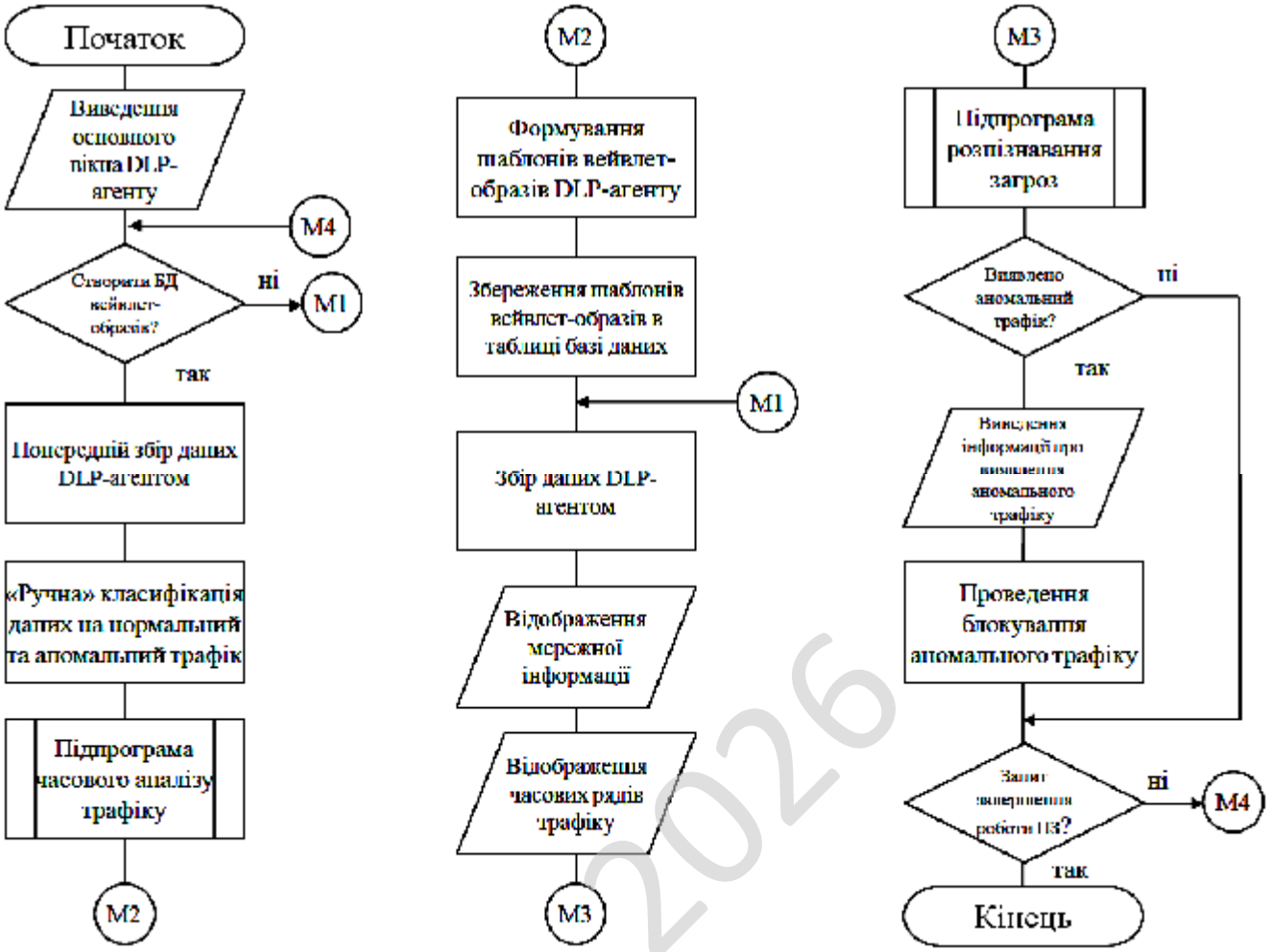

Рисунок 4.1 – Блок-схема основної програми

Але вже до кінця 1980-х років існуюча тоді реляційна модель перестала задовольняти розробників в силу низки обмежень. Відповіддю на зростаючу складність програм баз даних стали два нових напрямки розвитку СКБД: об'єктно-орієнтовані СКБД і об'єктно-реляційні СКБД.

У 1991 був утворений консорціум ODMG, основною метою якого стало вироблення промислового стандарту об'єктно-орієнтованих баз даних. Між 1993 та 2001 роками ODMG опублікувала п'ять ревізій своїх специфікацій. Остання версія стандарту має індекс 3.0, після чого група розпустилася. До кінця 1990-х існувало близько десяти компаній, що виробляли комерційні продукти, що позиціонуються на ринку як ООСКБД. Найбільш відомими системами даного класу стали Objectivity, Versant виробництва однойменних компаній, а також

СКБД Jasmine, випущена компанією CA. Незважаючи на переваги, що дозволяють ефективніше вирішувати певний ряд завдань, об'єктно-орієнтовані системи так і не змогли завоювати значущу частку ринку СКБД, залишившись «нішевим» продуктом.

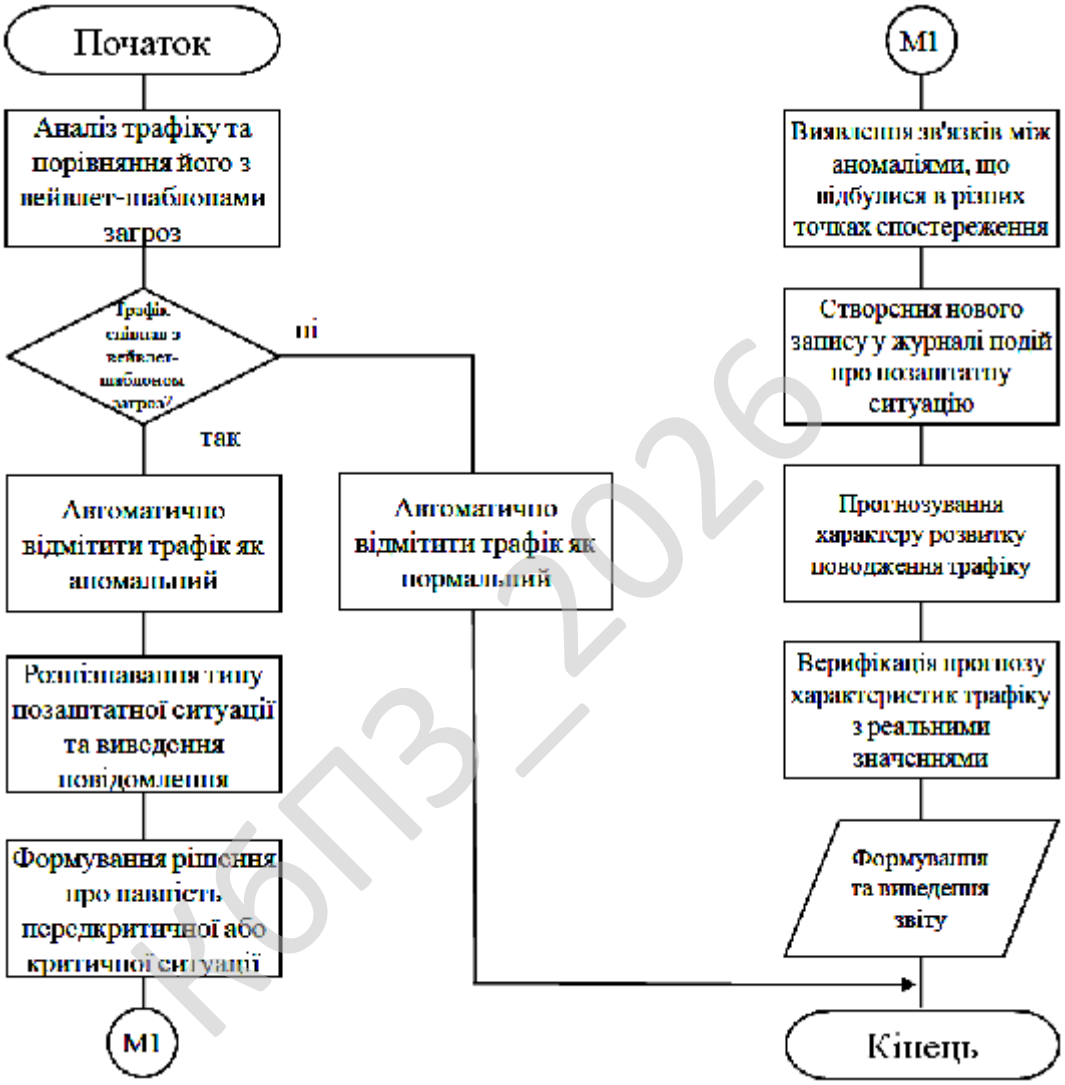

Рисунок 4.2 – Блок-схема роботи підпрограми

Постачальниками традиційних реляційних СКБД також була проведена значна робота з об'єднання об'єктно-орієнтованих і реляційних систем. Розробники постаралися розширити мову SQL, щоб включити в неї концепції об'єктно-орієнтованого підходу, зберігаючи переваги реляційної моделі (об'єктні розширення мови SQL були зафіксовані в стандарті SQL:1999).

Основний принцип – це еволюційний розвиток можливостей СКБД без поломки попередніх підходів та зі збереженням наступності з системами попереднього покоління.

Поняття СКБД третього покоління, якими, власне кажучи, і є об'єктно-реляційні СКБД, з'явилося після опублікування групою відомих фахівців в області баз даних «Маніфесту систем баз даних третього покоління». Основні принципи СКБД третього покоління, позначені в маніфесті:

Крім традиційних послуг з управління даними, СКБД третього покоління повинні забезпечити підтримку розвиненіших структур об'єктів і правил. Розвинутіша структура об'єктів характеризує засоби, необхідні для зберігання і маніпулювання нетрадиційними елементами даних (тексти, просторові дані, мультимедіа).

СКБД третього покоління повинні включити в себе СКБД другого покоління. Системи другого покоління внесли вирішальний вклад у двох областях – непроцедурний доступ за допомогою мови запитів SQL і незалежність даних. Ці досягнення обов'язково повинні враховуватися в системах третього покоління.

СКБД третього покоління повинні бути відкриті для інших підсистем. Це включає оснащення різноманітними інструментами підтримки прийняття рішень, доступом з багатьох мов програмування, інтерфейсами до існуючих популярних систем і бізнес-застосунків, можливістю запуску програм з бази даних на іншій машині і розподілені СКБД. Весь набір інструментів і СКБД має ефективно функціонувати на різноманітних апаратних платформах з різними операційними системами.

Крім того, СКБД, що розраховує на широку сферу застосування, повинна бути оснащена мовою четвертого покоління (4GL).

У середині 1990 років було лише кілька дослідних прототипів СКБД, які поєднали найкращі риси реляційних і об'єктно-орієнтованих СКБД. Першим комерційним продуктом, якому були властиві об'єктно-реляційні риси, став

Universal Server компанії Informix(згодом була поглинена IBM). В даний час більшість цих ідей вже втілено в реальних комерційних рішеннях, в тому числі і в продуктах основних постачальників СКБД (Oracle Database і IBM DB2).

Розвиток індустрії систем керування базами даних базується на значних фундаментальних наукових дослідженнях. Найчастіше, між самими дослідженнями та їхньою конкретною реалізацією в прикладних рішеннях минають роки, а іноді й десятиліття. Роботу в області управління даними проводять як університетські дослідницькі групи (MIT, Berkeley), так і центри розробок основних постачальників СКБД (Oracle, IBM, Microsoft). Інвестування в управління даними – це довгострокове, і разом з тим, вигідне вкладення коштів. В даний час дослідники мають у своєму розпорядженні засоби, що дозволяють ефективно реалізувати найскладніші запити, що маніпулюють терабайтами й петабайтами різних даних.

Основними тенденціями, які дали привід для проведення різних масштабних досліджень в області баз даних стали:

Експонентний ріст даних. Обсяг даних, у тому числі синтетичних, що генеруються автоматизованими системами, значно зріс. Збільшилося і число прикладних областей, в яких вимагається обробка великих обсягів даних. До таких областей тепер відносяться не тільки традиційні корпоративні програми, пошук у веб, але також і наукові дослідження, обробка природних мов, аналіз соціальних мереж тощо.

Значне ускладнення структур використовуваних даних. Прості види даних у вигляді чисел і символьних рядків стали доповняться численною мультимедійною інформацією, просторовими, процедурними даними та великою кількістю інших складних форматів.

Широке поширення дешевих високопродуктивних апаратних засобів. Щорічно ми спостерігаємо зростання обчислювальних можливостей мікропроцесорів, збільшення ємності і зниження вартості доступних і зручних в експлуатації пристроїв дискової оперативної пам'яті.

Активний розвиток засобів комунікації та «всесвітньої павутини» World Wide Web. WWW стає єдиним інформаційним середовищем, що пронизує весь світ і об'єднує величезне число користувачів та електронних пристроїв.

Поява нових важливих областей застосування СКБД. У першу чергу, це пов'язано з інтелектуальним аналізом даних, сховищами даних, а останнім часом – з паралельними обчисленнями і хмарними технологіями.

Основні характеристики СКБД

1. Контроль за надлишковістю даних.

2. Несуперечливість даних.

3. Підтримка цілісності бази даних (коректність та несуперечливість).

4. Цілісність описується за допомогою обмежень.

5. Незалежність прикладних програм від даних.

6. Спільне використання даних.

7. Підвищений рівень безпеки.

Можливості СКБД

1. Дозволяється створювати БД (здійснюється за допомогою мови визначення даних DDL (Data Definition Language)).

2. Дозволяється додавання, оновлення, видалення та читання інформації з БД (за допомогою мови маніпулювання даними DML, яку часто називають мовою запитів)

3. Можна надавати контрольований доступ до БД за допомогою: Системи забезпечення захисту, яка запобігає несанкціонованому доступу до БД; Системи керування паралельною роботою прикладних програм, яка контролює процеси спільного доступу до БД; Система відновлення – дозволяє відновлювати БД до попереднього несуперечливого стану, що був порушений в результаті збою апаратного або програмного забезпечення.

Основні компоненти середовища СКБД

1. Апаратне забезпечення.

2. Програмне забезпечення.

3. Дані.

4. Процедури – інструкції та правила, які повинні враховуватись при проектуванні та використанні БД.

5. Користувачі: адміністратори даних (керування даними, проектування БД, розробка алгоритмів, процедур) та БД (фізичне проектування, відповідальність за безпеку та цілісність даних); розробники БД; прикладні програмісти; кінцеві користувачі.

Архітектура СКБД

Існує трирівнева система організації СКБД ANSI-SPARC, при якій існує незалежний рівень для ізоляції програми від особливостей представлення даних на нижчому рівні.

Рівні:

1. Зовнішній – представлення БД з точки зору користувача.

2. Концептуальний – узагальнене представлення БД, описує які дані зберігаються в БД і зв'язки між ними. Підтримує зовнішні представлення, підтримується внутрішнім рівнем.

3. Внутрішній – фізичне представлення БД в комп'ютері.

Логічна незалежність – повна захищеність зовнішніх моделей від змін, що вносяться в концептуальну модель.

Фізична незалежність – захищеність концептуальної моделі від змін, які вносяться у внутрішню модель.

Опис системи

Програмне забезпечення системи реалізації інтелектуального DLP агента працює як локальний модуль контролю витоку даних. Система аналізує текстові файли, службові повідомлення та вихідні дані користувача, визначає наявність конфіденційної інформації, розраховує рівень ризику та приймає рішення про дозвіл, блокування або передавання події до журналу безпеки.

Архітектура системи складається з модуля збору даних, модуля пошуку чутливих шаблонів, модуля інтелектуальної оцінки ризику, модуля політик

безпеки, модуля журналювання та модуля формування звітів. Основна логіка реалізується мовою Python, а події зберігаються у базі SQLite.

Основні функції системи полягають у виявленні персональних даних, номерів банківських карток, службових ключів доступу, паролів, внутрішніх документів, розрахунку ризику за кількістю знайдених ознак, блокуванні небезпечних операцій, збереженні доказової інформації та формуванні короткого звіту для адміністратора.

Система працює в режимі агента на робочій станції. Вона не передає вміст документів у відкритому вигляді, а зберігає тільки хеш файлу, тип інциденту, рівень ризику та рішення політики. Це підвищує безпеку та зменшує обсяг конфіденційних даних у журналі.

Наведемо частину коду, яка реалізую деякі з вищеперерахованих функцій.

```
import re
import sqlite3
import hashlib
from dataclasses import dataclass
from datetime import datetime
from pathlib import Path

@dataclass
class DLPEvent:
    source: str
    content_hash: str
    detected_types: list
    risk_score: int
    decision: str
    created_at: str
class SensitivePatternEngine:
    def __init__(self):
        self.patterns = {
            "банківська картка": r"\b(?:\d[ -]*?){13,16}\b",
            "email адреса": r"\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-
z]{2,}\b",
            "токен доступу":
r"\b(?:api_key|token|secret|access_key)\s*[=]\s*[A-Za-z0-9_\-]{12,}\b",
            "пароль": r"\b(?:password|passwd|pwd)\s*[=]\s*[^ \n]{6,}\b",
            "службовий документ": r"\b(confidential|internal use
```

```
only|комерційна таємниця|конфіденційно)\b"
            }
        def detect(self, text):
            found = []
            for name, pattern in self.patterns.items():
                if re.search(pattern, text, flags=re.IGNORECASE):
                    found.append(name)
            return found
    class RiskAnalyzer:
        def calculate(self, detected_types, text):
            score = 0
            score += len(detected_types) * 25
            if len(text) > 5000:
                score += 10
            if "комерційна таємниця" in text.lower():
                score += 20
            if "confidential" in text.lower():
                score += 20
            return min(score, 100)

    class PolicyEngine:
        def decide(self, risk_score):
            if risk_score >= 70:
                return "заблоковано"

            if risk_score >= 40:
                return "потрібна перевірка адміністратора"

            return "дозволено"


    class AuditStorage:
        def __init__(self, db_path="dlp_agent.db"):
            self.db_path = db_path
            self._create_tables()

        def _create_tables(self):
            with sqlite3.connect(self.db_path) as connection:
                connection.execute("""
                    CREATE TABLE IF NOT EXISTS dlp_events (
                        id INTEGER PRIMARY KEY AUTOINCREMENT,
                        source TEXT NOT NULL,
```

```
                    content_hash TEXT NOT NULL,
                    detected_types TEXT NOT NULL,
                    risk_score INTEGER NOT NULL,
                    decision TEXT NOT NULL,
                    created_at TEXT NOT NULL
                )
            """)

    def save_event(self, event):
        with sqlite3.connect(self.db_path) as connection:
            connection.execute("""
                INSERT INTO dlp_events (
                    source,
                    content_hash,
                    detected_types,
                    risk_score,
                    decision,
                    created_at
                )
                VALUES (?, ?, ?, ?, ?, ?)
            """, (
                event.source,
                event.content_hash,
                ", ".join(event.detected_types),
                event.risk_score,
                event.decision,
                event.created_at
            ))

    def get_report(self):
        with sqlite3.connect(self.db_path) as connection:
            rows = connection.execute("""
                SELECT decision, COUNT(*), AVG(risk_score)
                FROM dlp_events
                GROUP BY decision
            """).fetchall()

            return rows


class IntelligentDLPAgent:
    def __init__(self):
```

```
            self.pattern_engine = SensitivePatternEngine()
            self.risk_analyzer = RiskAnalyzer()
            self.policy_engine = PolicyEngine()
            self.storage = AuditStorage()

        def analyze_text(self, source, text):
    #Коментар описує створення хешу замість збереження повного конфіденційного
тексту
            content_hash = hashlib.sha256(text.encode("utf-8")).hexdigest()

            detected_types = self.pattern_engine.detect(text)
            risk_score = self.risk_analyzer.calculate(detected_types, text)
            decision = self.policy_engine.decide(risk_score)

            event = DLPEvent(
                source=source,
                content_hash=content_hash,
                detected_types=detected_types,
                risk_score=risk_score,
                decision=decision,
                created_at=datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            )

            self.storage.save_event(event)
            return event

        def analyze_file(self, file_path):
            path = Path(file_path)

            if not path.exists():
                raise FileNotFoundError("Файл не знайдено")

            text = path.read_text(encoding="utf-8", errors="ignore")
            return self.analyze_text(str(path), text)

        def print_report(self):
            report = self.storage.get_report()

            print("Звіт DLP агента")
            for decision, count, avg_score in report:
                print(f"Рішення {decision}. Кількість {count}. Середній ризик
{avg_score:.2f}")
```

```
def main():
    agent = IntelligentDLPAgent()

    sample_text = """
    Документ конфіденційно.
    Email користувача admin@example.com
    password=SuperSecret123
    api_key=ABCD1234567890SECRET
    """

    event = agent.analyze_text("demo_document.txt", sample_text)

    print("Результат аналізу")
    print(f"Джерело {event.source}")
    print(f"Виявлені ознаки {', '.join(event.detected_types)}")
    print(f"Ризик {event.risk_score}")
    print(f"Рішення {event.decision}")

    agent.print_report()


if __name__ == "__main__":
    main()
```

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

Було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу).

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею 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.

Діаграма прецедентів це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

– асоціації (association relationship);

– включення (include relationship);

– розширення (extend relationship);

– узагальнення (generalization relationship).

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SEED – у криптографії симетричний блоковий криптоалгоритм на основі Мережі Фейстеля, розроблений Корейським агентством інформаційної безпеки (Korean Information Security Agency, KISA) в

1998 році. В алгоритмі використовується 128-бітний блок і ключ довжиною 128 біт. Алгоритм одержав широке поширення й використовується фінансовими й банківськими структурами, виробничими підприємствами й бюджетними установами Південної Кореї, оскільки 40-бітний SSL не забезпечує на даний момент мінімально необхідного рівня безпеки. Агентством по захисту інформації специфіковане використання шифру SEED у протоколах TLS і S/MIME. У той же час, алгоритм SEED не реалізований у більшості сучасних браузерів і інтернет-додатків, що утрудняє його використання в даній сфері поза межами Південної Кореї.

SEED являє собою мережу Фейстеля з 16 раундами, 128-бітовими блоками й 128-бітовим ключем. Алгоритм використовує дві 8×8 таблиці підстановки, які, як такі з Safer, виведені з дискретного зведення в ступінь (у цьому випадку, x^{247} і x^{251} – плюс деякі «несумісні операції»). Це є деякою подібністю с MISTY1 у рекурсивності його структури: 128-бітовий повний шифр – мережа Фейстеля з F-функцією, що впливає на 64-бітові половини, у той час як сама F-функція – Мережа Фейстеля, складена з G-функції, що впливає на 32-розрядні половини. Однак рекурсія не простягнеться далі, тому що G-функція – не Мережа Фейстеля. В G-функції 32-розрядне слово розглядають як чотири 8-бітових байта, кожний з яких проходить через одну або іншу таблицю підстановки, потім поєднується в помірковано комплексному наборі булевих функцій таким чином, що кожний біт виводу залежить від 3 з 4 вхідних байтів.

SEED має складний ключовий розклад, генеруючи тридцять два 32-розрядних додаткових символу, використовуючи G-функції на серіях обертань вихідного неопрацьованого ключа, комбінованого зі спеціальними раундовими константами (як в TEA) від «Золотого співвідношення» (англ. Golden ratio).

Згідно з дослідженнями KISA, алгоритм SEED «надійно протистоїть відомим атакам».

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі програмне забезпечення системи реалізації DLP-агенту. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

– Верхнього меню: Файл; Журнал подій; Параметри; Довідка.

– Розділу обрання групи обробки.

– Розділу виведення результату роботи системи.

– Функції представлені у графічному вигляді.

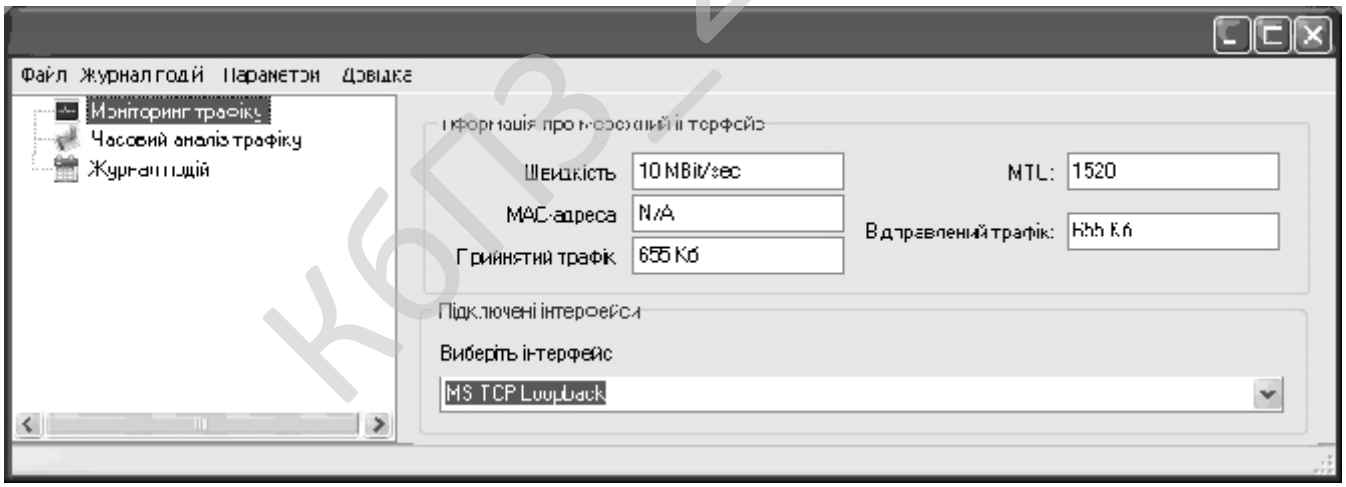

Рисунок 5.1 – Головне вікно ПЗ

Повномасштабне впровадження DLP-системи складно представити без моніторингу кінцевих точок. Обмеження на використання USB-носіїв, контроль буфера обміну – ці канали виходить контролювати тільки на стороні агента.

Якщо ж говорити про комунікаційний трафік, то сучасні технології дозволяють перехоплювати майже все на мережному шлюзі й проксі-сервері. Навіть аналізувати SSL-трафік у прозорому режимі без налаштувань у властивостях браузера. Тому ми рекомендуємо завжди розумно підходити до вибору перехоплювачів. Занадто часте рішення «простіше на агенті» виявляється глибоко помилковим.

Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

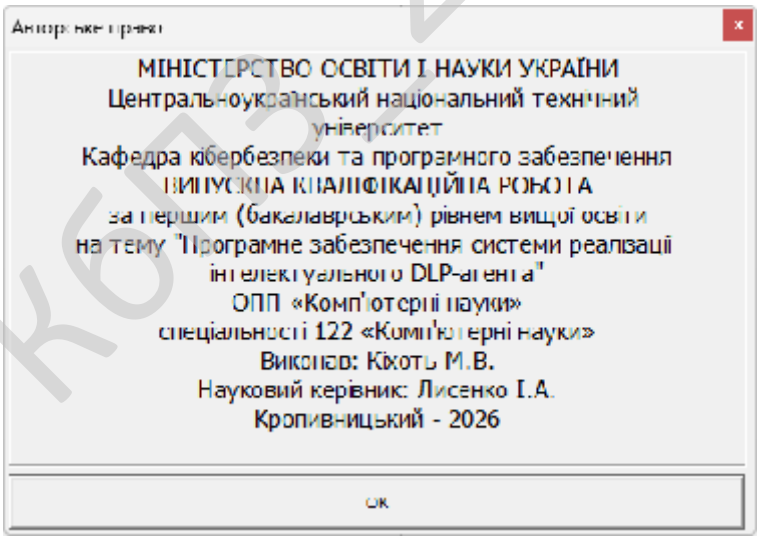

Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження

програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

– Випуск.

– Встановлення та активація.

– Деактивація.

– Адаптація.

– Обновлення.

– Вмонтування.

– Відстежування версій.

– Видалення.

– Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження.

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в IT рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

– відповідності поставленим вимогам;

– правильна відповідь для усіх можливих вхідних даних;

– виконання функцій за прийнятний час;

– практичність;

– сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Обрано умови розповсюдження – Freeware. Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільності як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи реалізації інтелектуального DLP-агента.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем реалізації інтелектуального DLP-агента.

– Досліджена система реалізації інтелектуального DLP-агента.

– На основі отриманих результатів досліджень створена програмна реалізація системи реалізації інтелектуального DLP-агента.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання реалізації інтелектуального DLP-агента.

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

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

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи реалізації інтелектуального DLP-агента. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне

програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм SEED.

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

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Vijay Kumar Velu. Mastering Kali Linux for Advanced Penetration Testing. Packt Publishing Ltd. 2022. 573 p.

2. Josh Armitage. Cloud Native Security Cookbook. O'Reilly Media. 2022. 516 p.

3. Massimo Bertaccini. Cryptography Algorithms. Packt Publishing. 2022. 358 p.

4. Alyssa Miller. Cybersecurity Career Guide. Manning Publications. 2022. 368 p.

5. Awais Rashid, Howard Chivers, George Danezis, Emil Lupu, Andrew Martin. CyBOK The Cyber Security Body of Knowledge. The National Cyber Security Centre. 2019. 854 p.

6. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.

7. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.

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

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

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

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

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

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

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

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

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

17. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев, О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки банківських систем з використанням штучного інтелекту». *Кібербезпека: освіта, наука, техніка.* 2025. Том 1 № 29. С.704–716, 2025

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

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

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

21. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with

Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023*, 2025. vol 389. pp 377-389. Springer, Singapore.

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

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

24. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447.

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

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

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

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

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

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

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

32. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings,* Volume 3530, 2023, pp. 256-265.

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

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

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

36. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *ІІ Міжнародна науково-практична Інтернет-конференція «Інновації та*

перспективні шляхи hозвитку інформаційних технологій (ІПШРІТ-2023)» м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

59. 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.

60. 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.