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 інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків”

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

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

Бєльченку Владиславу Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків*

2. Керівник роботи *Коваленко Олександр Володимирович, докт. техн. наук, професор*
(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 інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків*

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
_____ Коваленко О.В.
(прізвище та ініціали)

Завдання прийнято до виконання
« 6 » лютого 2026 р.

Підпис здобувача
_____ Бєльченко В.С.
(прізвище та ініціали)

АНОТАЦІЯ

Бєльченко В.С. Програмне забезпечення системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків. 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 інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

Метою розробки є програмне забезпечення системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

Результат роботи – програмна реалізація системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

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, цифрові водяні знаки

ABSTRACT

Bielchenko V.S. Software for the intellectual copyright system using digital watermarks. 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 intellectual copyright system using digital watermarks.

The purpose of the development is software for the intellectual copyright system using digital watermarks.

The result of the work is the software implementation of the intellectual copyright system using digital watermarks.

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

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

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Python environment.

Keywords: computer science, digital watermarking

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ВДТ	–	відео-дисплейні термінали
ЕОМ	–	електронно-обчислювальна машина
ЕЦП	–	електронний цифровий підпис
ПЗ	–	програмне забезпечення
ПК	–	персональний комп’ютер
СБ	–	служба безпеки
ТЗ	–	технічне завдання
ЦВЗ	–	цифрові водяні знаки
DES	–	стандарт шифрування США
DSA	–	Digital Signature Algorithm
ECDSA	–	Elliptic Curve Digital Signature Algorithm
EGSA	–	El Gamal Signature Algorithm
IDEA	–	International Date Encryption Algorithm – алгоритм шифрування
IP	–	Internet Protocol
LSB	–	Least Significant Bits – метод стеганографії
PGP	–	Pretty Good Privacy – міжнародний криптографічний стандарт
RSA	–	алгоритм асиметричного шифрування
SHA-1	–	Secure Hash Algorithm 1 – алгоритм криптографічного гешування
TDES	–	Triple DES –модифікація DES з трьома незалежними підключами
JPEG	–	Joint Photographic Experts Group – растровий формат зображення

ВСТУП

Актуальність теми. Цифрові водяні знаки – це техніка, яка використовується для вбудовування непомітної інформації в цифрові носії, такі як зображення, відео, аудіо та документи, з метою захисту права власності, відстеження несанкціонованого розповсюдження та перевірки автентичності. На відміну від видимих водяних знаків, які навмисно накладаються на контент, цифрові водяні знаки розроблені таким чином, щоб залишатися невидимими, але водночас їх можна було виявити спеціалізованим програмним забезпеченням.

Оскільки створення цифрового контенту прискорюється завдяки генерації за допомогою штучного інтелекту, хмарній співпраці та глобальному розповсюдженню, цифрові водяні знаки стали фундаментальною технологією захисту контенту. У 2026 році організації покладаються на цифрові водяні знаки не лише для забезпечення дотримання авторських прав, але й для судово-медичного виявлення витоків, забезпечення прозорості за допомогою штучного інтелекту та довіри до бренду.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

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

1. Огляд існуючих систем інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

2. Дослідження системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

3. Програмна реалізація системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків, 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

Такі компанії, як Digimarc та Nagra, здобули свою репутацію в епоху, коли водяні знаки в основному асоціювалися з телевізійним мовленням та боротьбою з піратством. Їхні технології чудово зарекомендували себе в контрольованих робочих процесах, де медіафайли рухалися через відомі канали розповсюдження. Але сучасне медіасередовище більше не контролюється. Контент тепер миттєво поширюється між платформами соціальних мереж, інструментами генерації штучного інтелекту, хмарними середовищами редагування, додатками для обміну повідомленнями, екосистемами для творців та спільними корпоративними платформами.

Одне зображення чи відео можна зробити з екрана, стиснути, відредагувати, обрізати, повторно відрендерити, опублікувати повторно або передати через кілька систем штучного інтелекту за лічені хвилини. Традиційні системи водяних знаків мають труднощі в цьому середовищі з кількох причин:

– Багато хто покладається на детерміновані або засновані на правилах методи вбудовування, які не були розроблені для того, щоб витримувати агресивні трансформації штучного інтелекту.

– Системи виявлення часто дають збій після зміни розміру, повторного стиснення або генеративних модифікацій.

– Старіші рішення були оптимізовані для медіакомпаній та мовників, а не для сучасної корпоративної співпраці чи екосистем штучного інтелекту.

– Інтеграція в сучасні робочі процеси для творців залишається громіздкою.

– Вони ніколи не були розроблені для походження синтетичних медіа чи перевірки контенту, згенерованого штучним інтелектом.

У NAB цей розрив був очевидним. Галузь більше не запитує, чи може водяний знак саме по собі зупинити піратство. Галузь запитує, чи може водяний знак встановити довіру у світі, де будь-хто може створювати фотореалістичні медіафайли за лічені секунди.

1.2 Область застосування

Зростання генеративного штучного інтелекту повністю змінило очікування. Нанесення водяних знаків – це вже не просто відстеження витоків після їх виникнення. Воно дедалі частіше стосується відповіді на набагато важливіше питання: «Чи можна довіряти цим ЗМІ?» Саме через цей зсув технології водяних знаків на основі машинного навчання так швидко набирають обертів.

На відміну від старіших систем, які часто спиралися на методи статичного вбудовування, сучасні платформи водяних знаків на основі штучного інтелекту спеціально навчаються витримувати трансформації, маніпуляції та синтетичні робочі процеси.

Сам водяний знак стає адаптивним та стійким, оскільки моделі машинного навчання можуть вивчати, як медіа змінюється в різних середовищах. Ця різниця має величезне значення. У NAB багато постачальників та керівників ЗМІ відкрито обговорювали занепокоєння щодо:

4. Дезінформація, створена штучним інтелектом.
5. Синтетичні актори та голоси.
6. Несанкціоноване маніпулювання контентом.
7. Крадіжка інтелектуальної власності.
8. Витік медіафайлів перед релізом.
9. Перевірка автентичності для журналістики.
10. Перевірка ланцюга зберігання цифрових активів.

Традиційні постачальники водяних знаків просто не вели ці розмови з таким самим рівнем довіри чи інновацій. Натомість значна частина енергії була зосереджена на новіших підходах компаній, створених спеціально для ери штучного інтелекту.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків, 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

Наведені нижче інструменти представляють одні з найпотужніших рішень для цифрових водяних знаків, доступних сьогодні, кожне з яких задовольняє різні потреби.

Steg.AI

Steg.AI – це платформа, орієнтована на підприємства, яка пропонує передові засоби судово-медичного цифрового нанесення водяних знаків на зображення, відео, аудіо та документи. Її технологія розроблена для вбудовування непомітних ідентифікаторів, які витримують поширені трансформації, такі як стиснення, зміна розміру та редагування, що робить її особливо ефективною для виявлення витоків та атрибуції. Steg.AI широко використовується компаніями у сфері медіа та розваг, побутової електроніки та іграшок, яким потрібно відстежувати, як конфіденційний контент переміщується між внутрішніми командами, партнерами або загальнодоступним Інтернетом. Платформа надає як веб-інтерфейс, так і доступ до API, що дозволяє вбудовувати водяні знаки безпосередньо у виробничі робочі процеси. Хоча її орієнтація на підприємства означає, що ціноутворення та налаштування можуть перевищувати потреби звичайних користувачів, вона добре підходить для команд, яким потрібне масштабоване, автоматизоване та високостійке цифрове нанесення водяних знаків.

Синтетичний ідентифікатор Google

Google SynthID – це система цифрових водяних знаків, розроблена Google DeepMind для ідентифікації контенту, згенерованого штучним інтелектом,

включаючи зображення, аудіо та текст. Замість того, щоб зосереджуватися на традиційних випадках використання для виявлення авторських прав або витоків, SynthID розроблений для задоволення зростаючої потреби в прозорості навколо генеративних результатів ШІ. Водяний знак вбудований непомітно та призначений для того, щоб залишатися виявленим навіть після звичайних трансформацій. SynthID відображає ширше прагнення галузі до відповідального розгортання ШІ та визначення походження контенту. Однак, його сфера застосування обмежена, оскільки він застосовується лише до контенту, згенерованого ШІ, і наразі не позиціонується як універсальне рішення для цифрових водяних знаків або судово-медичного аналізу підприємств.

Цифрові водяні знаки Digimarc

Digimarc є одним із найвідоміших постачальників у сфері цифрових водяних знаків, що пропонує високонадійні рішення для автентифікації медіа, захисту брендів та застосувань у ланцюгах поставок. Його технологія широко використовується великими підприємствами, яким потрібна висока точність виявлення та довготривала надійність як цифрових, так і фізичних активів. Рішення Digimarc для водяних знаків відомі своєю стійкістю та масштабованістю, але зазвичай вони передбачають складні процеси адаптації, ліцензування та інтеграції. Як результат, Digimarc найкраще підходить для великих організацій з вимогами до захисту контенту для дорослих, а не для стартапів чи окремих творців.

OpenStego

OpenStego – це безкоштовний інструмент з відкритим вихідним кодом, орієнтований на базову стеганографію та вбудовування водяних знаків. Він часто використовується для освітніх цілей, експериментів або невеликих особистих проектів, а не для комерційного розгортання. Хоча OpenStego може вбудовувати приховану інформацію в зображення, йому бракує надійності, автоматизації та можливостей криміналістики, необхідних для сучасного захисту контенту у великих масштабах. Немає вбудованої інфраструктури відстеження витоків,

виявлення або стійкості до складних перетворень, що обмежує його застосування в професійному середовищі. Тим не менш, він залишається корисною відправною точкою для розуміння основ цифрового водяного знаку.

Студія водяних знаків Arclab

Arclab Watermark Studio розроблено в першу чергу для фотографів та творчих професіоналів, які хочуть наносити видимі або прості невидимі водяні знаки на зображення. Програмне забезпечення підкреслює простоту використання та пакетну обробку, що робить його придатним для брендингу та маркування авторських прав. Однак воно не призначене для судово-медичного цифрового нанесення водяних знаків, визначення атрибуції витоків або випадків використання в корпоративній безпеці. Його функціональність обмежена зображеннями, і воно не забезпечує можливостей виявлення або відстеження після розповсюдження контенту. Як результат, його найкраще використовувати для базової сигналізації про авторські права, а не для надійного захисту контенту.

Вибір правильного інструменту для цифрових водяних знаків

Вибір правильного рішення для цифрового водяного знаку у 2026 році залежить від проблеми, яку ви намагаєтеся вирішити. Корпоративні команди, що спеціалізуються на виявленні витоків та атрибуції, тяжітимуть до таких платформ, як Steg.AI або Digimarc. Організації, що займаються ідентифікацією контенту, створеного штучним інтелектом, можуть вважати Google SynthID більш підходящим. Творчі професіонали та окремі користувачі можуть віддати перевагу простішим інструментам, таким як Arclab Watermark Studio, тоді як дослідники та аматори можуть експериментувати з варіантами з відкритим кодом, такими як OpenStego.

Оскільки цифровий контент продовжує масштабуватися, а медіа, створене штучним інтелектом, стає все більш поширеним, цифрові водяні знаки залишатимуться критично важливою технологією для захисту інтелектуальної власності, забезпечення автентичності та підтримки довіри в цифрових екосистемах.

Чому Steg.AI стає галузевим стандартом

Серед найбільш обговорюваних технологій була Steg.AI, чия платформа для судово-медичного нанесення водяних знаків на основі машинного навчання, схоже, дедалі більше відповідає напрямку розвитку галузі. Особливо привабливим Steg.AI робить те, що компанія була створена з урахуванням сучасних медіа-реалій, а не застарілих припущень щодо мовлення. Його технологія нанесення водяних знаків розроблена для того, щоб:

– Непомітний для людського ока.

– Стійкий до маніпуляцій.

– Зберігається протягом усіх трансформацій.

– Масштабований для зображень, відео, аудіо та документів.

– Інтегровано в існуючі робочі процеси за допомогою API та плагінів.

– Сумісний з ініціативами щодо походження та автентичності, такими як C2PA.

Це не просто водяний знак для запобігання піратству. Це водяний знак для майбутнього, де кожен медіафайл може потребувати вбудованих метаданих довіри. Одним із найпомітніших зрушень у NAB стало те, як часто учасники обговорювали «судово-медичне нанесення водяних знаків», а не просто «нанесення водяних знаків».

Ця відмінність має значення. Судово-медичне нанесення водяних знаків передбачає атрибуцію, відстеження та підзвітність. Це дозволяє організаціям визначати, звідки походить контент, хто мав до нього доступ і, можливо, як він розповсюджувався. У галузях, що мають справу з ембаргованими медіафайлами, конфіденційними активами, неопублікованим розважальним контентом або конфіденційними корпоративними документами, ця можливість стає важливою. Традиційних видимих водяних знаків більше недостатньо. Сучасні організації хочуть невидимих, стійких до штучного інтелекту систем атрибуції, які не погіршують взаємодію з користувачем.

Схоже, що Steg.AI позиціонує себе безпосередньо на цій кривій попиту.

SynthID від Google сигналізує про напрямок розвитку галузі

Мабуть, найпереконливішим доказом того, що галузь відходить від застарілих підходів до водяних знаків, стала зростаюча увага до SynthID з боку Google. Коли одна з найбільших у світі компаній зі штучного інтелекту значно інвестує в технології водяних знаків та визначення походження, вона надсилає ринку сигнал: інфраструктура довіри стає основоположною. SynthID являє собою щось більше, ніж просто черговий інструмент для додавання водяних знаків. Він являє собою стратегічне визнання того, що медіа, створені за допомогою штучного інтелекту, потребують вбудованих систем автентифікації з моменту створення контенту. Це масштабний філософський зсув порівняно зі старими парадигмами водяних знаків. Застарілі системи зазвичай сприймали водяні знаки як щось, що додається пізніше в робочому процесі – часто під час розповсюдження або постпродакшну. SynthID та подібні системи, що використовують штучний інтелект, натомість інтегрують автентичність безпосередньо в процес створення контенту. Такий підхід набагато масштабованіший для майбутнього генеративних медіа. У NAB багато розмов щодо безпеки контенту на основі штучного інтелекту стосувалися систем походження, стандартів автентичності та вбудованих рівнів довіри. Участь Google ефективно прискорила галузеву перевірку водяних знаків як критичного рівня управління штучним інтелектом. І що важливо, це підштовхнуло ринок до підходів, заснованих на машинному навчанні.

Галузевий стандарт змінюється в режимі реального часу

У NAB ставало дедалі очевиднішим те, що галузевий стандарт більше не визначається виключно традиційними гравцями. Компанії, які ведуть сьогоднішні дискусії, вирішують проблеми завтрашнього дня. Це включає:

– Автентифікація медіа, згенерована штучним інтелектом.

– Відстеження походження.

– Пом'якшення дипфейків.

– Запобігання витоку корпоративного контенту.

– Збереження атрибуції.

– Управління синтетичним контентом.

– Міжплатформна медіа-довіра.

Постачальники традиційних продуктів досі підтримують міцні корпоративні зв'язки та багаторічний операційний досвід. Вони не зникають за одну ніч, але багато хто з них зараз, схоже, реагують на зміни, а не керують ними. Тим часом, такі компанії, як Steg.AI, та технології, як SynthID, відчувають свою відповідність напрямку активного руху ринку. Енергія навколо цих платформ у NAB відображала щось ширше, що відбувається в галузі: водяні знаки еволюціонують з нішевої функції безпеки в ключовий компонент інфраструктури цифрової довіри. Ця трансформація змінює все.

Дивлячись у майбутнє

Наступні п'ять років, ймовірно, визначатимуть, які компанії стануть постачальниками базової інфраструктури для екосистеми штучного інтелекту для медіа. Майбутнє водяних знаків належатиме не компаніям, які просто захищають трансляції. Воно належатиме компаніям, які можуть встановити автентичність, атрибуцію, походження та довіру в інтернеті, який дедалі більше переповнений штучним контентом. Ось чому підходи на основі машинного навчання так швидко набирають популярності. І саме тому такі технології, як Steg.AI та SynthID, дедалі менше схожі на конкурентів, що тільки починають виходити на ринок, а радше на початок нового галузевого стандарту.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Як мова програмування обрана Python. Python – високорівнева мова програмування загального призначення з акцентом на продуктивність розроблювача й читаність коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує кілька парадигм програмування, у тому числі структурне, об'єктно-орієнтоване, функціональне, імперативне й аспектно-орієнтоване. Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточні обчислень і зручні високорівневі структури даних. Код у Python організовується у функції й класи, які можуть поєднуватися в модулі (які у свою чергу можуть бути об'єднані в пакети).

Еталонною реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ. Він поширюється вільно під дуже ліберальною ліцензією, що дозволяє використовувати його без обмежень у будь-яких застосунках, включаючи пропрієтарні. Є реалізації інтерпретаторів для JVM (з можливістю компіляції), MSIL (з можливістю компіляції), LLVM і інших. Проект PyPy пропонує реалізацію Python на самому Python, що зменшує витрати на зміни мови й постановку експериментів над новими можливостями.

Python – мова програмування, що активно розвивається, нові версії (з додаванням/зміною язикових властивостей) виходять приблизно раз у два з половиною року. Внаслідок цього й деяких інших причин на Python відсутні ANSI, ISO або інші офіційні стандарти, їхня роль виконує CPython.

Python портований і працює майже на всіх відомих платформах – від КПК до мейнфреймів. Існують порти під Microsoft Windows, практично всі варіанти UNIX (включаючи FreeBSD і Linux), Plan 9, Mac OS і Mac OS X, iPhone OS 2.0 і вище, Palm OS, OS/2, Amiga, AS/400 і навіть OS/390, Symbian і Android.

При цьому, на відміну від багатьох портуємих систем, для всіх основних платформ Python має підтримку характерних для даної платформи технологій (наприклад, Microsoft COM/DCOM). Більше того, існує спеціальна версія Python для віртуальної машини Java – Jython, що дозволяє інтерпретаторові виконуватися на будь-якій системі, що підтримує Java, при цьому класи Java можуть безпосередньо використовуватися з Python й навіть бути написаними на

Python. Також кілька проектів забезпечують інтеграцію із платформою Microsoft .NET, основні з яких – IronPython і Python.Net.

Python підтримує динамічну типізацію, тобто тип змінної визначається тільки під час виконання. Тому замість «присвоювання значення змінної» краще говорити про «зв'язування значення з деяким ім'ям». У Python є убудовані типи: бульові, рядки, Unicode-рядки, цілі числа довільної точності, числа із плаваючої коми, комплексні числа й деякі інші. З колекцій Python підтримує кортежі (*tuples*), списки, словники (асоціативні масиви) і, починаючи з версії 2.4, безлічі. Всі значення в Python є об'єктами, у тому числі функції, методи, модулі, класи.

Додати новий тип можна або написавши клас (class), або визначивши новий тип у модулі розширення (наприклад, написаному мовою C). Система класів підтримує спадкування (одиночне й множинне) і метапрограмування. Можливе спадкування від більшості убудованих типів і типів розширень.

Всі об'єкти діляться на посилальні й атомарні. До атомарного ставляться int, long, complex і деякі інші. При присвоюванні атомарних об'єктів копіюється їхнє значення, у той час як для посилальних копіюється тільки покажчик на об'єкт, таким чином, обидві змінні після присвоювання використовують те саме значення. Посилальні об'єкти бувають змінювані й незмінні. Наприклад, рядки й кортежі є незмінними, а списки, словники й багато інших об'єктів – змінюваними. Кортеж у Python є, по суті, незмінним списком. У багатьох випадках кортежі працюють швидше списків, тому якщо ви не плануєте змінювати послідовність, то краще використовувати саме їх. Мова має чіткий і послідовний синтаксис, продуману модульність й масштабованість, завдяки чому вихідний код написаних на Python програм легко читаємий. Python – стабільна й розповсюджена мова. Він використовується в багатьох проектах і в різних якостях: як основна мова програмування або для створення розширень і інтеграції застосунків. На Python реалізоване велика кількість проектів, також він активно використовується для створення прототипів майбутніх програм. Python використовується в багатьох великих компаніях.

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 інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

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

В цифрову епоху зміни – єдина константа. Орієнтуючись у цьому постійно мінливому ландшафті, ви неодмінно зіткнетеся з подвійною силою цифрової трансформації та прав інтелектуальної власності. Ці два елементи дедалі більше переплітаються, формуючи те, як ми створюємо, поширюємо та захищаємо цифровий контент.

Цифрова трансформація, з її обіцянкою оптимізації процесів та підвищення ефективності, революціонізує галузі. Однак вона не позбавлена своїх викликів. Одним із таких викликів є захист прав інтелектуальної власності, критичний, але часто недооцінений аспект.

У цьому розділі ми заглибимося у складний взаємозв'язок між цифровою трансформацією та правами інтелектуальної власності. Ми дослідимо, як вони впливають одне на одного та чому вам варто звернути на це увагу. Тож, незалежно від того, чи ви цифровий підприємець, ентузіаст технологій чи просто допитливий, ця стаття обіцяє бути повчальним чтивом.

Огляд цифрової трансформації

Цифрова трансформація являє собою кардинальну зміну в сучасному бізнес-ландшафті. З огляду на безперервні технологічні інновації, поширені в кожній галузі, розуміння цифрової трансформації охоплює основні бізнес-функції, відносини з клієнтами та, зокрема, права інтелектуальної власності (ІВ).

Цифрова трансформація, що складається з низки змін, спричинених інтеграцією цифрових технологій в усі сфери бізнесу, суттєво змінює те, як ви працюєте та надаєте цінність клієнтам. Колись домінували паперові сліди та ручні процеси, тепер замінені цифровими робочими процесами, аналітикою даних та розумнішими системами.

Цей перехід полягає не лише у заміні простоїв на інновації. Йдеться про впровадження цифрових технологій, які змінюють структуру вашої діяльності, змушуючи вас переосмислити традиційні бізнес-моделі. Чи то підвищення продуктивності за допомогою цифрових інструментів, чи впровадження більш гнучкого робочого середовища, цифрова трансформація надає вам конкурентну перевагу на ринку.

У процесі цифрової трансформації компанії все частіше розробляють або використовують програмні додатки, аналітичні інструменти, цифровий контент, технологічні платформи тощо для підвищення ефективності свого бізнесу. Як наслідок, обсяг інтелектуальної власності (ІВ), патентів, авторських прав, торгових марок, що належать компаніям, також суттєво зростає. Забезпечення відповідних прав інтелектуальної власності для таких цифрових активів стає центральним питанням, що робить інтелектуальну власність невід'ємним аспектом цифрової трансформації.

Розуміння цього взаємозв'язку цифрової трансформації та прав інтелектуальної власності в цифрову епоху є життєво важливим. Воно забезпечує вашому бізнесу ясність, необхідну для управління новими цифровими викликами. У наступних розділах ми заглибимося в практичні аспекти та приклади цього взаємозв'язку, щоб дати вам всебічне розуміння предмета.

Права інтелектуальної власності в цифрову епоху

Розвиток цифрових технологій змінює середовище для прав інтелектуальної власності (ІВ). Першочерговий фокус захисту ІВ зазнає численних викликів, оскільки технології продовжують свій невпинний рух. Постійно нові цифрові активи та необхідність їх захисту поставили права ІВ у центр уваги в цій технологічній еволюції.

У боротьбі з цифровою трансформацією права інтелектуальної власності стають ключовим інструментом. Наприклад, патенти зберігають ексклюзивність цифрових винаходів, надаючи бізнесу конкурентну перевагу. Торгові марки, як ще один аспект прав інтелектуальної власності, забезпечують ідентичність

бренду та репутацію у віртуальному просторі.

Однак процес отримання цифрових прав інтелектуальної власності – це не проста прогулянка віртуальним парком. Врахуйте, що закони про авторське право захищають твори мистецтва; цифрові творіння, такі як програмне забезпечення, електронні книги, музика, можна легко копіювати та поширювати через Інтернет. Оригінальні творці можуть зіткнутися зі значними труднощами, якщо не з прямими збитками.

Крім того, передові інструменти, такі як алгоритми пошукових систем та аналітика даних, ускладнюють захист інтелектуальної власності. Підприємства можуть використовувати їх для відстеження тенденцій та оптимізації своїх стратегій, але їхні віртуальні ідентичності можуть стати вразливими. Тому вкрай важливо забезпечити відповідність використання цих інструментів законодавству про інтелектуальну власність.

Більше того, цифровий фронтир приносить низку нових викликів у сфері інтелектуальної власності. Інновації в технології доповненої або віртуальної реальності, хмарних сервісах та блокчейні пропонують новаторські способи створення та розповсюдження контенту. Отже, цифрова трансформація зумовлює потребу в динамічних стратегіях захисту прав інтелектуальної власності.

Цифрова ера є переконливим аргументом на користь збалансованого підходу до прав інтелектуальної власності. Закони та політика у сфері інтелектуальної власності повинні адаптуватися до цифрового переходу. Вони повинні забезпечувати еволюцію цифрового ландшафту, одночасно захищаючи права особистості та заохочуючи інновації. Бізнес також повинен залишатися пильним, щоб ефективно долати виклики, пов'язані з правами інтелектуальної власності, в епоху цифрової трансформації. Йдеться вже не про те, «чи» цифрова трансформація вплине на права інтелектуальної власності, а про те, «як» і якою мірою. Зрештою, досягнення цифрової ери залишаться з нами.

Вплив цифрової трансформації на права інтелектуальної власності

Цифрова трансформація створює численні виклики для прав

інтелектуальної власності в цифрову епоху. Одним із критичних викликів є швидкий технологічний прогрес, який постійно перевизначає параметри прав інтелектуальної власності.

По-перше, оскільки цифрова трансформація сприяє інноваціям у більших масштабах, відокремлення оригінальних дизайнів від тиражованих стає дедалі складнішим. Наприклад, у випадку розробки програмного забезпечення, розмежування натхненних концепцій та скопійованих фрагментів коду створює унікальні труднощі. Як наслідок, такі компанії, як ваша, стикаються з труднощами у виявленні порушень та захисті своїх цифрових активів.

По-друге, розгляньте, як цифрові технології дозволяють легко дублювати та розповсюджувати творчі роботи. Припустімо, ви маєте справу з цифровим контентом, таким як музика, книги чи фільми. У такому разі їхня цифрова природа робить їх легкодоступними та легко відтворюваними, що підвищує ризики несанкціонованого дублювання та розповсюдження. Більше того, притягнення винних до відповідальності є складним завданням через глобальний характер Інтернету.

По-третє, поява трансформаційних інструментів та платформ, таких як штучний інтелект, блокчейн та хмарні обчислення, породжує нові дискусії щодо інтелектуальної власності. Наприклад, якщо штучний інтелект створює витвір мистецтва, хто володіє авторським правом? Аналогічно, оскільки блокчейн забезпечує безпечну мережу для цифрових транзакцій, виникають питання щодо патентних прав та ліцензування. Нарешті, хмарні сервіси пропонують такі переваги, як гнучкий доступ, але вони ускладнюють питання щодо того, хто володіє та контролює дані.

Зрештою, перетин технологій та права інтелектуальної власності призводить до постійно мінливого правового ландшафту. Оскільки цифрова трансформація формує нові галузі та руйнує старі, вона змушує до постійної еволюції правової практики. Наприклад, судам та законодавцям довелося переглянути принципи авторського права, коли 3D-друк сприяв виготовленню

копій об'єктів, захищених авторським правом.

Пам’ятайте, що розуміння цих наслідків цифрової трансформації на права інтелектуальної власності потрібне не лише ентузіастам юриста. Як власнику бізнесу, який орієнтується в цифрову епоху, це життєво важливо для вашого виживання в умовах технологічних потрясінь. Будьте проактивними у розумінні змін меж прав інтелектуальної власності та розробляйте свої стратегії; від цього може залежати успіх вашого бізнесу.

Тематичні дослідження цифрової трансформації та прав інтелектуальної власності

Проаналізуємо кілька прикладів, які підкреслюють зв'язок між цифровою трансформацією та правами інтелектуальної власності в цифрову епоху.

Приклад 1: Потокове передавання онлайн-контенту та права інтелектуальної власності

Безпрецедентний розвиток технологій призвів до зростання популярності платформ онлайн-стрімінгу контенту, таких як Netflix, Amazon Prime та Disney+. Однак цифрова трансформація викликала проблеми порушення авторських прав. Наприклад, Netflix, титан онлайн-стрімінгу, зіткнувся з позовом у 2021 році від «ChooseCo» через використання терміну «обери свою власну пригоду» в одному з епізодів серіалу «Чорне дзеркало: Брандашмиг». Незважаючи на позов, Netflix продовжує збагачувати користувацький досвід, надаючи інтерактивні функції, пам’ятаючи про права інтелектуальної власності та вживаючи заходів для запобігання порушенню авторських прав.

Приклад 2: Штучний інтелект (ШІ) та інтелектуальна власність

Вплив штучного інтелекту на права інтелектуальної власності є помітним. Мовна модель OpenAI, GPT-3, створює текст, подібний до людського, на основі наданих даних. Нещодавно використання GPT-3 створило проблему з авторським правом, коли його використовували для відтворення статей з «The New York Times» без явного дозволу. Цей інцидент підкреслює складність, з якою стикаються підприємства під час інтеграції інструментів цифрової трансформації,

таких як ШІ, що вимагає чіткого розуміння прав інтелектуальної власності.

Приклад 3: Цифрове мистецтво та NFT

Ринок цифрового мистецтва та невзаємозамінних токенів (NFT), що розвивається, породжує нові питання, пов'язані з правами інтелектуальної власності. У березні 2021 року цифровий витвір мистецтва «Щоденні дні: Перші 5000 днів» художника Біпла був проданий на аукціоні Christie's як NFT за надзвичайну суму в 69 мільйонів доларів. Продаж стимулював дискусії щодо оригінальності цифрового мистецтва та потенційних порушень інтелектуальної власності. Цифрова трансформація, у цьому прикладі, виводить мистецтво в новий вимір, що робить критично важливим для цифрових художників розуміння та повагу до прав інтелектуальної власності.

Ці тематичні дослідження ілюструють виклики, що виникають внаслідок цифрової трансформації з точки зору захисту прав інтелектуальної власності в цифрову епоху. Вони підкреслюють нагальну та постійну потребу для бізнесу залишатися на крок попереду, розуміючи та дотримуючись законів про інтелектуальну власність в умовах швидкого технологічного прогресу.

Стратегії зміцнення прав інтелектуальної власності в цифрову епоху

Оскільки вплив цифрової трансформації продовжує зростати, важливо вживати проактивних заходів для зміцнення прав інтелектуальної власності (ІВ). Ось ключові стратегії, на яких слід зосередитися:

1. Навчання зацікавлених сторін: Підвищуйте обізнаність серед своїх співробітників, партнерів та чітко викладіть правила щодо прав інтелектуальної власності. Це зменшує порушення та проблеми, спричинені незнанням. Належне навчання обмежує кількість порушень та підкреслює важливість етики в цифровому просторі.

2. Використовуйте технології: використання передових технологій може посилити захист прав інтелектуальної власності. Передові алгоритми та технологія блокчейн забезпечують безпечніші засоби зберігання цифрових активів та гарантують перевірку автентичності.

3. Контролюйте онлайн-активність: регулярно відстежуйте свій цифровий слід. Різні інструменти допомагають кількісно визначити несанкціоноване використання вашої інтелектуальної власності, зокрема цифрові водяні знаки та системи зворотного пошуку зображень.

4. Впроваджуйте суворі заходи безпеки: кібербезпека не може бути другорядною думкою в цифрову епоху. Захистіть свої цифрові активи, постійно оновлюючи протоколи безпеки, використовуючи інструменти шифрування та маючи надійну стратегію резервного копіювання та аварійного відновлення.

5. Виступайте за сильніше законодавство: Лобіюйте за прийняття надійних та актуальних законів про авторське право, які враховують цифровий перехід. Цілеспрямовані, точні та всеохоплюючі правила можуть мінімізувати правові сірі зони в цифровій сфері.

6. Зверніться за юридичною консультацією: звернення за юридичною консультацією зменшує ризик ненавмисного порушення прав і забезпечує краще розуміння того, як захистити свої активи в умовах швидкозмінного цифрового середовища.

Пам’ятайте, що прийняття цифрової трансформації не означає відмову від контролю над правами інтелектуальної власності. Розуміння викликів та впровадження ефективних стратегій допомагають у збереженні прав, що, своєю чергою, сприяє креативності та інноваціям у цифровому просторі. Це посилення безпосередньо сприяє зростанню бізнесу. Зрештою, пам’ятайте, що ваші права інтелектуальної власності залишаються невід’ємною частиною ідентичності вашого бізнесу. Ретельно захищайте їх у цю цифрову епоху.

Майбутнє прав інтелектуальної власності в умовах цифрової трансформації

Уявіть собі світ, де цифрова трансформація продовжує прискореними темпами змінювати парадигми, порушуючи традиційні системи інтелектуальної власності (ІВ). Права інтелектуальної власності стають рухомою ціллю, складним завданням, яке стає ще складнішим з розвитком передових технологій, таких як

штучний інтелект, технології розподіленого реєстру, такі як блокчейн, та передові формати цифрових медіа.

У цю цифрову епоху новатори та власники бізнесу, ймовірно, боротимуться з трансформаційними технологіями, розробляючи нові підходи до управління та захисту своїх прав інтелектуальної власності. Ключовими особливостями цієї майбутньої сфери можуть бути розпізнавання контенту, захищеного авторським правом, у режимі реального часу та інструменти аналізу патентів на базі штучного інтелекту.

Наприклад, уявіть собі вплив блокчейну, відомого тим, що він пропонує безпечне, прозоре та незмінне ведення обліку. Застосування цієї технології в управлінні цифровими правами може революціонізувати виявлення порушень авторських прав, забезпечуючи незаперечні докази у разі судового розгляду.

Крім того, варто замислитися над тим, як цифрова трансформація може змінити законодавство щодо штучного інтелекту та його результатів. Оскільки штучний інтелект продовжує розвиватися, цілком можливо, що він може досягти точки, коли самостійно створюватиме оригінальний контент. Отже, питання права власності на такий контент може стати предметом юридичних дебатів, що потенційно може спонукати до внесення змін до законодавства для кращої адаптації до цих технічних досягнень.

Більше того, в умовах мінливого середовища глобальний охоплення Інтернету яскраво підкреслює необхідність міжконтинентальної узгодженості в законодавстві про цифрову інтелектуальну власність, сприяючи гармонійному регулюванню та механізмам правозастосування в різних юрисдикціях.

Зрештою, досягнення правильного балансу в цю цифрову епоху, динамічний простір, що характеризується поєднанням нових технологій, зміненої поведінки користувачів та регуляторних заходів, що розвиваються, залишається вирішальним для забезпечення постійного захисту прав інтелектуальної власності в умовах цифрових трансформацій. Оскільки цифрові технології все глибше проникають у наше життя, то й витонченість та широта наших законів про

інтелектуальну власність також повинні йти за цим прикладом. Застосування такого передбачення вимагає адаптивності, пильності та наполегливості. Пам’ятайте, що йдеться про проактивність, а не про реактивність у ваших стратегіях.

Отже, ви зорієнтувалися у складному ландшафті цифрової трансформації та прав інтелектуальної власності. Зрозуміло, що цифрова епоха приносить як виклики, так і можливості для прав інтелектуальної власності. Ви бачили, як передові інструменти можуть розмити межі між оригінальним та тиражованим контентом, і як штучний інтелект, блокчейн та хмарні обчислення змінюють дискусії. Ви також дослідили тематичні дослідження від потокових платформ до цифрового мистецтва та NFT, підкреслюючи нагальну потребу в надійних стратегіях захисту. Очевидно, що освіта, технології, онлайн-моніторинг, сувора безпека, сильніше законодавство та юридичні консультації є ключем до зміцнення прав інтелектуальної власності. Заглядаючи в майбутнє, ви зіткнетеся з новими викликами, пов'язаними з новими технологіями, а також з потенційними рішеннями, такими як інструменти розпізнавання авторських прав у режимі реального часу. Пам'ятайте, що оскільки штучний інтелект наближається до створення оригінального контенту, узгоджені глобальні закони стають ще більш важливими. У цьому постійно мінливому цифровому ландшафті ваша здатність адаптуватися та захищати свої права інтелектуальної власності є важливішою, ніж будь-коли. Бізнес стикається зі зростаючими труднощами у захисті торговельних марок, оскільки штучний інтелект змінює правила гри. Незважаючи на ефективність систем EUIPO та WIPO, традиційні механізми часто не справляються з новими викликами. У цій статті розглядається вплив штучного інтелекту на захист торговельних марок та практичні пропозиції для бізнесу щодо захисту своїх брендів. Штучний інтелект використовується недобросовісними продавцями для створення підроблених зображень товарів, оглядів та веб-сайтів, які можуть імітувати відомі бренди. Те саме стосується зловживання в рекламі, коли використання ШІ може ввести споживачів в оману та створити хибну

комерційну асоціацію. Закон ЄС про ШІ (2024/1689) містить положення, спрямовані на боротьбу з імітацією та обманом за допомогою ШІ. Стаття 50 Закону запроваджує нові зобов'язання щодо прозорості: будь-яке зображення, відео чи аудіо, які можуть сприйматися як справжні (діпфейк), повинні бути чітко позначені як штучні стороною, яка їх використовує. Очікується, що ці правила набудуть повної чинності у 2025-2026 роках.

Порівняльні регуляторні підходи: Велика Британія та США

Наприклад, у Великій Британії ASA активно обмежує дипфейкову рекламу, а суди визнають порушення під час використання брендів у цифрових рекламних кампаніях. Тим часом у Сполучених Штатах регулювання в цій сфері базується на Законі Ленхема та правилах FTC, які поширюються на фейкові асоціації та недобросовісну рекламу.

Як навчання ШІ викликає занепокоєння щодо авторського права?

Також проблематичним є той факт, що ШІ використовує юридично захищені дані під час своєї роботи та навчання, як-от у справі Getty Images проти Stability AI, в якій Getty стверджує, що Stability AI скопіював понад 12 мільйонів фотографій з колекції Getty Images разом із відповідними підписами та метаданими без дозволу Getty Images або компенсації Getty Images у рамках своїх зусиль щодо створення конкуруючого бізнесу. Водночас, штучний інтелект може також допомогти захистити торговельні марки, наприклад, за допомогою інструментів моніторингу в режимі реального часу та автентифікації продуктів для швидшого припинення порушень.

Необхідність проактивної стратегії

Постачальники моделей штучного інтелекту зобов'язані гарантувати, що їхні системи не генерують контент, який порушує права третіх сторін, включаючи торговельні марки. Однак, поки що такі заходи мають обмежену ефективність – звідси й потреба в правових та технічних гарантіях для бізнесу. Компанії більше не можуть покладатися лише на реєстрацію та пасивне правозастосування; їм потрібні проактивні та багаторівневі стратегії для

запобігання довгостроковій шкоді, такій як розмивання бренду, фінансові втрати, підрив довіри споживачів та шахрайські транзакції.

Що можуть зробити підприємства для захисту торгових марок від штучного інтелекту?

Реєстрація та управління торговою маркою:

– Реєстрація торгової марки через EUIPO (для ЄС), Мадридську систему BOIB (міжнародно) або через національні відомства. Без реєстрації правовий захист неможливий.

– Вкрай важливо постійно оновлювати реєстрацію: поновлювати період захисту, вчасно оновлювати дані для ребрендингу та реєструвати зміни до назви, логотипу та інших ідентифікаторів.

– Належний правовий захист має бути забезпечений шляхом реєстрації знака у відповідних класах товарів та послуг згідно з Ніццькою класифікацією (НКК). За необхідності розширте охоплення на пов'язані класи, особливо в цифровому середовищі (наприклад, програмне забезпечення, реклама, медіапослуги), де можуть здійснюватися спроби використання вашого бренду.

– Цифрові інструменти (EUIPO User Area, TM view, eMadrid, Madrid Monitor) слід використовувати для відстеження оновлень статусу, отримання сповіщень про потенційні конфлікти та актуальних даних щодо вашої реєстрації в режимі реального часу.

Інструменти моніторингу:

– Cloudflare Bot Management допомагає контролювати та блокувати автоматизований збір даних, включаючи сканери штучного інтелекту.

– Порушення прав третіх сторін (включаючи візуальні копії) можна відстежувати за допомогою бази даних брендів BOIB, Мадридського моніторингу та спеціалізованих інструментів штучного інтелекту.

– Захист контенту посилено водяними знаками, цифровими відбитками та вбудованими метаданими для ідентифікації джерела у разі виникнення суперечки.

– Файли Robots.txt та протокол REP можуть обмежувати парсинг, але вони працюють лише за умови добровільного дотримання.

– Щоб обмежити TDM (інтелектуальний аналіз тексту та даних), додайте чітку заборону на використання контенту в навчанні зі штучного інтелекту до умов використання сайту та вкажіть обмеження ліцензії. Технічні токени (наприклад, метадані) можуть доповнювати захист.

Механізми правозастосування в цифровому просторі:

– Інфраструктурні заходи, такі як автоматичне блокування неавторизованих ботів, дозволяють обмежити навчання ШІ вашого контенту та збір даних з вашого сайту без дозволу.

– Порушення прав, включаючи візуальні та семантичні копії торгової марки, можна виявити за допомогою систем розпізнавання, таких як eSearch plus, TMview та Віденський помічник з класифікації EUIPO.

– Заборона на використання контенту для цілей штучного інтелекту повинна бути чітко зазначена в користувацьких угодах, ліцензійних угодах та умовах використання онлайн-платформ.

– Якщо виявлено неправомірне використання торгової марки, на цифрові платформи слід надсилати повідомлення про видалення контенту, що порушує авторські права, через адміністративні або добровільні процедури.

– У разі систематичних або комерційно значних порушень можна застосувати статтю 53(1)(c) Регламенту (ЄС) 2024/1689 (Закон про штучний інтелект), яка зобов'язує постачальників моделей загального призначення дотримуватися законодавства ЄС про авторське право, включаючи визнання та поважання правових застережень, висловлених відповідно до статті 4(3) Директиви (ЄС) 2019/790 (Директива про CDSM).

Захист торгових марок в епоху штучного інтелекту

Компанії повинні реагувати на стрімко змінюване середовище; лише ручне забезпечення дотримання правил більше неможливе в сучасних умовах. Щоб залишатися ефективними, стратегії захисту торговельних марок повинні

включати інструменти штучного інтелекту та адаптуватися до нових технологічних реалій. Як і будь-яка потужна технологія, вона вимагає постійного нагляду та свідомого, кваліфікованого використання.

Хоча ШІ має трансформаційний потенціал, його розвиток не повинен відбуватися за рахунок прав інтелектуальної власності. Правовим системам потрібен час для адаптації до нових технологій, і нові виклики продовжуватимуть виникати. Ці проблеми повинні вирішуватися не лише урядами та регуляторами, а й завдяки активній участі та відповідальності самих компаній. Досягнення сталого балансу між законними інноваціями та незаконним привласненням залежатиме як від правового захисту, так і від активних зусиль власників брендів щодо створення безпечнішого середовища, керованого штучним інтелектом.

3.2 Розробка структурної схеми

Структурна схема системи у вигляді сукупності схеми пристроїв стеганографічного приховування та вилучення даних в просторовій області зображень із використанням прямого розширення спектру, які побудовані за пропонованим способом зображено на рис. 3.1.

Пристрій стеганографічного приховування даних (див. рис. 3.1) працює наступним чином. Джерело інформаційних повідомлень формує послідовність інформаційних даних, які подаються пристрій шифрування, ініційований таємним ключем K_3, що формується джерелом ключів 3. Зашифровані інформаційні повідомлення подаються на пристрій перешкодостійкого кодування, в якому виконується внесення спеціально формованої надмірності для підвищення достовірності інформаційних зашифрованих даних. Джерело ключів 2 формує таємний ключ K_2, який ініціює генератор псевдовипадкових послідовностей. Результатом роботи генератору псевдовипадкових послідовностей є дискретні сигнали, тобто дискретні послідовності, елементи яких сформовано псевдовипадковим чином. Сформовані псевдовипадкові

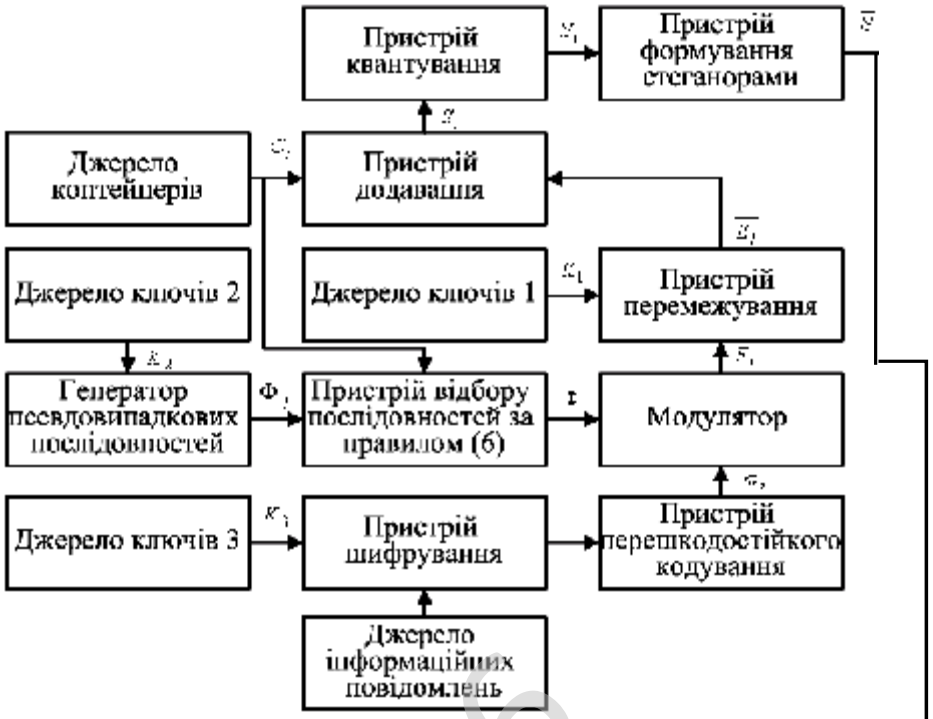
послідовності Φ_j поступають на додатково (в порівнянні із відомим способом) введений пристрій відбору послідовностей, в якому для всіх $i = 0,..,N-1$ розраховується значення коефіцієнту кореляції

$$\rho\left(C_i,\Phi_j\right)=\frac{1}{n}\sum_{z=0}^{n-1}C_{i_z}\varphi_{j_z}$$

та порівнюється із наперед визначеним значенням $\rho_{\max}$.

У випадку, коли хоча б для одного $i \in \{0,..,N-1\}$ розраховане значення $\rho\left(C_i,\Phi_j\right)$ перевищить значення порогу $\rho_{\max}$ сформована псевдовипадкова послідовність бракується, тобто дискретні сигнали Φ_j із $\rho\left(C_i,\Phi_j\right) > \rho_{\max}$ хоча б для одного $i \in \{0,..,N-1\}$ для стеганографічного приховування інформаційних даних не застосовуються.

Якщо для сформованого дискретного сигналу Φ_j та для всіх $i = 0,..,N-1$ розраховані значення коефіцієнту кореляції $\rho\left(C_i,\Phi_j\right)$ менші або дорівнюють встановленому порогу $\rho_{\max}$, тобто, якщо виконується умова (3.14) для всіх блоків даних контейнеру, відповідне значення Φ_j приймається до подальшого стеганографічного приховування інформаційних даних. Сформовані таким чином псевдовипадкові послідовності складають ансамбль дискретних сигналів $\Phi = \{\Phi_0,\Phi_1,...,\Phi_{M-1}\}$, вони враховують статистичні властивості контейнера та подаються до модулятора. На модулятор подається також блок інформаційних даних $m_i = \left(m_{i_0},m_{i_1},...,m_{i_{k-1}}\right)$, $k \le M$, який модулюється псевдовипадковими послідовностями за правилом. Сформоване таким чином модульоване повідомлення E_i подається на пристрій перемежування, ініційованого таємним ключем K_1, який сформовано джерелом ключів 1.



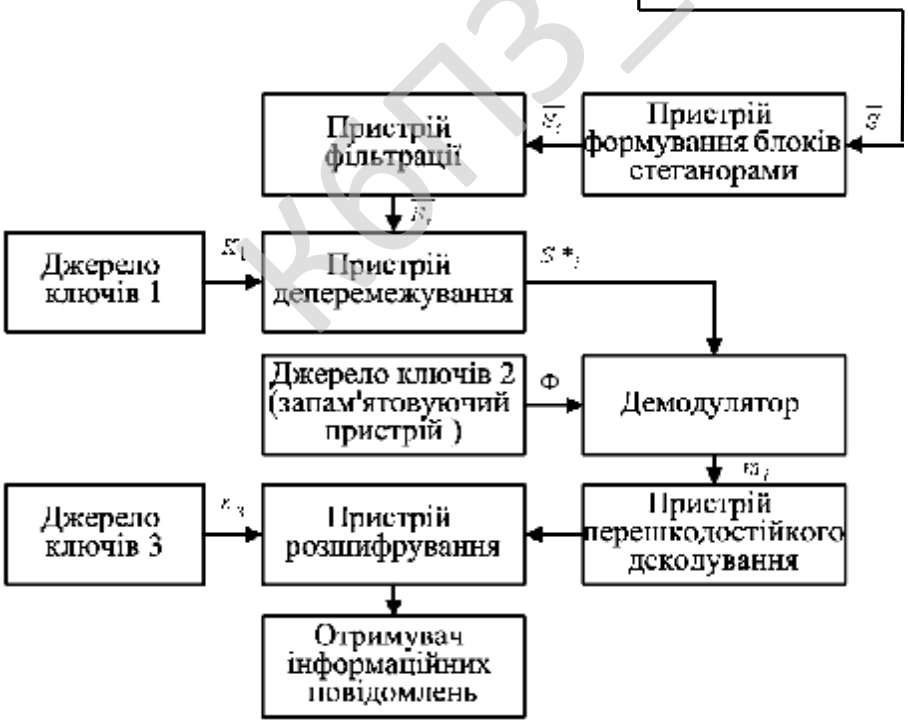


Рисунок 3.1 – Структурна схема системи

Пристрій перемежування обробляє модульоване повідомлення E_i, тобто за правилом f, яке задає таємний ключ K_1, псевдовипадковим чином переставляє місцями елементи E_i. Отримані дані $\overline{E_i} = f(E_i, K_1)$ подаються на пристрій додавання, у якому виконується поелементне додавання з даними контейнеру C_i (з даними цифрового зображення в просторовій області): $S_i = C_i + \overline{E_i} \cdot G$, де $G > 0$ – коефіцієнт підсилення розширювального сигналу, який задає «енергію» вбудованих блоків інформаційного повідомлення. Контейнер формується джерелом контейнерів. Отримані дані S_i подаються на пристрій квантування, який виконує певне перетворення для зберігання початкового динамічного діапазону зображення-контейнеру, в результаті чого формуються окремі блоки стеганограми $\overline{S_i}$ та заповнений контейнер $\overline{S} = \overline{S_0} \cup \overline{S_1} \cup ... \cup \overline{S_{N-1}}$, який передається приймальній стороні.

Пристрій стеганографічного вилучення даних (див. рис. 3.1) працює аналогічно відповідному пристрою як і у відомому способі [1-3], за винятком генератора псевдовипадкових послідовностей, що формує відповідні дискретні сигнали. Пристрій функціонує наступним чином. Отримана стеганограмма $\overline{S}$ на приймальній стороні подається на пристрій формування блоків стеганограми, в якому формуються блоки $\overline{S_i}$ та подаються на пристрій фільтрації. Після фільтрації отримані дані $\overline{\overline{S_i}}$ подаються на пристрій зворотного перемежування, на якому виконується дія, інверсна перемежуванню на передавальній стороні. Пристрій деперемежування ініційовано секретним ключем K_1, який сформовано джерелом ключів 1. Отримані після деперемежування дані S^*_i подаються на демодулятор, який виконує функцію кореляційного приймача дискретних сигналів за розглянутим вище правилом. Тобто в демодуляторі обчислюється значення коефіцієнту кореляції даних S^*_i та псевдовипадкових послідовностей Φ_i (дискретних сигналів). Ансамбль дискретних сигналів не формується (як у відомому способі) відповідним генератором псевдовипадкових чисел, що ініційований секретним ключем K_2, а зберігається у запам'ятовуючому

пристрою, тобто весь ансамбль псевдовипадкових послідовностей $\Phi = \{\Phi_0, \Phi_1, ..., \Phi_{M-1}\}$ виступає у ролі секретного ключа K_2 і зберігається цілком в такому вигляді. Таким чином, запам'ятовуючий пристрій можна розглядати як аналог джерела ключів 2 у відомому способі [3], а послідовності, які надходять з нього є тотожними тим, які застосовуються на передавальній стороні при вбудовуванні інформаційних повідомлень. Таким чином, в демодуляторі обчислюється значення коефіцієнту кореляції між отриманими даними S^*_i та послідовностями, які застосовувалися при вбудовуванні інформації. Рішення, стосовно значення вбудованих даних, приймається відповідно до значення обрахованого коефіцієнту кореляції за правилом (5). Вилучені дані m_i подаються на пристрій перешкодостійкого декодування, в якому за визначеним правилом із використанням внесеної надмірності виправляються деякі помилки, відповідно до корегуючої здатності коду. Це призводить до деякого підвищення достовірності переданих даних, які після декодування подаються на пристрій розшифрування, ініційованого таємним ключем K_3, що формується джерелом ключів 3. Розшифровані повідомлення подаються отримувачу інформаційних повідомлень.

Переваги водяних знаків на основі машинного навчання

1. Мотивація: чому «навчання» змінює проблему водяних знаків

Цифрове водяне маркування завжди балансувало між трьома конкуруючими цілями: непомітністю (мінімальна втрата якості), стійкістю (витримує трансформації) та ємністю (скільки інформації можна вбудувати). Машинне навчання (ML) змінює те, як ми задовольняємо цей компроміс: замість того, щоб розробляти фіксовані правила вбудовування вручну, ми вивчаємо процедури вбудовування та виявлення безпосередньо з даних та змодельованих «спотворень каналу» (наприклад, стиснення, масштабування, обрізання). Цей зсув дозволяє створювати системи водяних знаків, які є більш адаптивними до хаотичних, неоднорідних перетворень контенту в реальних дистрибутивних конвеєрах. Канонічним прикладом є HiDDeN, який розглядає водяні знаки як наскрізну навчальну систему, що складається з кодера (вбудовування), декодера

(відновлення) та диференційованого шару шуму/атаки (імітація спотворень під час навчання).

2. Надійність завдяки навчанню на реалістичних спотвореннях

Центральною емпіричною перевагою водяних знаків машинним навчанням є те, що стійкість можна навчати, а не просто проектувати. У HiDDeN стійкість покращується шляхом явної оптимізації точності декодування після проходження медіафайлів з водяними знаками через збурення під час навчання. Для відео RivaGAN аналогічно розглядає стійкість як мету навчання та вводить архітектурні варіанти (включаючи увагу) для підтримки відновлюваності водяних знаків за звичайних операцій обробки відео. В аудіо нещодавня робота стверджує, що заяви про стійкість повинні оцінюватися систематично, оскільки атаки сильно різняться; AudioMarkBench – один із таких бенчмарків, який тестує кілька методів водяних знаків за багатьох збурень та моделей загроз, ілюструючи як прогрес, так і прогалини, що залишилися.

3. Краща непомітність завдяки перцептивним цілям

Непомітність часто є тим, де підходи, засновані на навчанні, стають практично привабливими: моделі машинного навчання можуть оптимізуватися проти втрат сприйняття або інших цілей, які краще корелюють з людськими судженнями, ніж прості помилки на рівні пікселів або вибірок. Наприклад, AudioSeal явно навчає пару генератор/детектор і обговорює використання критеріїв сприйняття, одночасно орієнтуючись на стійкість до редагування та локалізації доказів водяного знака в довших аудіозаписах. Ширший сенс полягає в тому, що системи машинного навчання можуть інтерналізувати рішення, залежні від контенту, розміщуючи енергію водяного знака там, де вона менш помітна, але все ще декодується, а не рівномірно вбудовуючи її по всьому контенту.

4. Розпізнавання з урахуванням контенту та локалізації (корисно для відредагованих медіафайлів)

Частою помилкою в класичних системах є те, що часткове редагування (обрізання, сплайсинг, часткова заміна) може порушити виявлення. Помітний

напрямок роботи з машинного навчання намагається зробити виявлення локалізованим – виявляти «фрагменти» водяного знака в часі чи просторі – щоб водяний знак можна було виявити, навіть коли збереглася лише частина медіафайлу. AudioSeal позиціонується в цьому напрямку для контекстів клонування мовлення та голосу, підкреслюючи локалізоване виявлення під час редагування.

5. Масштабованість: практичне розгортання в масштабі платформи

Після навчання водяні знаки машинного навчання можуть бути операційно ефективними: вбудовування та виявлення можуть виконуватися автоматично та послідовно у великих обсягах контенту, з меншим налаштуванням кожного ресурсу, ніж у багатьох класичних конвеєрах. Саме на цьому підкреслює історію розгортання Steg.AI: судово-медичне водяне маркування здійснюється через API/інтеграції та позиціонується як стійке до поширених маніпуляцій та видалення метаданих. З прикладної точки зору це важливо, оскільки «система водяних знаків» рідко є просто алгоритмом; це робочий процес, інтегрований у виробництво, розповсюдження та реагування на інциденти контенту.

6. Більш точна судово-медична атрибуція (хто що вилив)

Для багатьох організацій водяні знаки – це не стільки доказ існування водяного знака, скільки атрибуція – пов’язування витоку ресурсу з певним одержувачем або подією розповсюдження. Steg.AI розглядає це як вбудовування унікальної ідентифікаційної інформації безпосередньо в медіа, призначеної для збереження після видалення метаданих та можливості відновлення після редагування. Підходи, засновані на навчанні, можуть підтримувати це шляхом вбудовування кодів з вищою надійністю та навчання детекторів для їх відновлення за реалістичних перетворень (і, все частіше, спроб видалення з боку конкурентів).

7. Адаптація до нових моделей загроз та медіа-каналів

Прагматичною перевагою водяних знаків машинного навчання є можливість перенавчання в міру розвитку стратегій атаки (наприклад, нові

кодеки, нові ланцюги доповнення або модифікації на основі штучного інтелекту). Оглядові роботи щодо стеганографії/водяних знаків на основі навчання підкреслюють, як ця галузь просунулася від ранніх глибоких фреймворків до більш надійних та різноманітних дизайнів, що свідчить про те, що ітеративне навчання та оцінка тепер є частиною життєвого циклу водяних знаків, а не одноразовим кроком проектування. Зокрема, в аудіо, бенчмаркінг та оцінки «системного стилю» підкреслюють, що надійність залежить від контексту та повинна підтримуватися в міру зміни інструментів генерації та редагування.

Технологія водяних знаків на основі машинного навчання є привабливою, оскільки перетворює її на комплексну задачу оптимізації: ви можете спільно навчитися, де приховувати інформацію, як зробити її непомітною та як відновлювати її після реалістичних спотворень. Академічні системи, такі як HiDDeN (зображення), RivaGAN (відео) та AudioSeal (аудіо), ілюструють цю траєкторію, тоді як Steg.AI представляє шлях комерціалізації, зосереджений на судово-медичному відстеженні та оперативному розгортанні.

3.3 Розробка функціональної схеми

Технологія застосування системи припускає наявність мережі абонентів, що посилають один одному підписані фотографії з мікропейментового фотобанку. Для кожного абонента генерується пара ключів: закритий і відкритий.

Закритий ключ зберігається абонентом в таємниці і використовується їм для формування ЕЦП.

Відкритий ключ відомий всім іншим користувачам мікропейментового фотобанку і призначений для перевірки ЕЦП одержувачем підписаної фотографії з мікропейментового фотобанку. Відкритий ключ не дозволяє обчислити секретний ключ. Кожен легальний користувач може за допомогою відкритого ключа перевірити достовірність і незмінність електронного документу.

Цифровий водяний знак, прихований у файлі, служить гарантом того, що навіть якщо зловмисник підбере закритий ключ і підпише файл, результати

перевірки підпису і ЦВЗ не співпадуть і можна буде встановити порушення. ЦВЗ виступає як додатковий рівень захисту, який іноді важко навіть виявити, а тим більше обійти.

Розроблена система складається з наступних модулів:

– генерація ключів (датчик псевдо-випадкових чисел);

– геш-функція SHA-1;

– криптографічний модуль (алгоритми RSA та DES);

– стеганографічний модуль (для вбудовування ЦВЗ).

Модуль генерації ключів створює для користувача закритий та відкритий RSA ключі, та надсилає запит на сертифікацію відкритого ключа. У разі успішного пройдешня сертифікації, відкритий ключ, термін його придатності та інформація про власника зберігаються в базі даних (репозиторії), для того, щоб одержувачі підписаних документів могли перевірити його автентичність.

Геш-функція та криптографічний модуль використовуються при створенні електронного цифрового підпису та при його перевірці. Також криптографічний модуль задіяно при шифруванні та дешифруванні документу. Асиметричний алгоритм RSA використовується для створення підпису, а блочний алгоритм DES для шифрування.

Стеганографічний модуль необхідний для вбудовування цифрового водяного знаку в графічне зображення відсканованого документу та для його вилучення з документу і перевірки.

Центр сертифікації ключів (англ. Certification authority, CA) – це організація, яка надає послуги електронного цифрового підпису, зокрема:

– Надання у користування засобів електронного цифрового підпису.

– Допомога при генерації відкритих та особистих ключів.

– Обслуговування сертифікатів ключів:

- формування;
- розповсюдження;
- скасування;

– зберігання;

– блокування;

– Надання інформації щодо чинних, скасованих і блокованих сертифікатів відкритих ключів.

– Послуги фіксування часу.

– Консультації та інші послуги.

Головним призначенням центру сертифікації є засвідчення автентичності відкритих ключів користувачів.

Центр сертифікації ключів, акредитований в установленому порядку, є акредитованим центром сертифікації ключів. Відмінністю акредитованого центру є те, що він має право обслуговувати виключно посилені сертифікати ключів.

Акредитований центр сертифікації ключів має виконувати усі зобов'язання та вимоги, встановлені законодавством для центру сертифікації ключів, та додатково зобов'язаний використовувати для надання послуг електронного цифрового підпису надійні засоби електронного цифрового підпису.

Порядок акредитації та вимоги, яким повинен відповідати акредитований центр сертифікації ключів, встановлюються Кабінетом Міністрів України.

Відкриті ключі й інша інформація про користувачів зберігається центрами сертифікації, у вигляді цифрових сертифікатів, що мають наступну структуру:

– серійний номер сертифіката;

– об'єктний ідентифікатор алгоритму електронного підпису;

– ім'я центру, що засвідчує;

– строк придатності;

– ім'я власника сертифіката (ім'я користувача, якому належить сертифікат);

– відкритий ключ власника сертифіката;

– об'єктні ідентифікатори алгоритмів, асоційованих з відкритими ключами власника сертифіката.

Рисунок 3.2 – Функціональна схема роботи системи

Система включає дві процедури:

– процедуру захисту документу цифровими водяними знаками, електронним підписом та шифруванням;

– процедуру розшифрування та перевірки легітимності документу, шляхом перевірки ЕЦП та ЦВЗ.

У процедурі створення підпису використовується закритий ключ RSA відправника, в процедурі перевірки підпису – відкритий ключ RSA відправника. При вбудовуванні ЦВЗ в документ застосовується стегоключ, а при шифруванні секретний DES ключ, щоб забезпечити секретність цих ключів, вони шифруються відкритим ключем RSA одержувача, після чого лише він може розшифрувати їх своїм закритим RSA ключем.

Принциповим моментом в системі ЕЦП є неможливість підробки електронного підпису користувача без знання його секретного ключа.

Кожен підпис містить наступну інформацію:

– дату підпису;

– термін закінчення дії ключа даного підпису;

– інформацію про особу, що підписала файл (П.І.Б., посада, коротке найменування фірми);

– ідентифікатор того, хто підписав (ім'я відкритого ключа);

– власне цифровий підпис.

Перед відправкою файлу по мережі, в нього вбудовується ЦВЗ. При формуванні ЕЦП відправник перш за все обчислює геш-функцію h(M) документу M, що слід підписати.

Обчислене значення геш-функції h(M) являє собою один короткий блок інформації m, що характеризує весь документ M в цілому. Потім число m шифрується секретним ключем відправника. Отримувана при цьому пара чисел є ЕЦП для даного документу M.

Далі документ шифрується і формується електронний конверт, що містить: зашифрований файл з попередньо вбудованими водяними знаками, підпис та зашифровані стего– та DES ключі. Електронний конверт передається по мережі, де може відбутися атака зловмисника на нього з метою підробки чи модифікації його змісту.

Прийнятий по каналу зв’язку документ M розшифровується, одержувач повідомлення, знову обчислює його геш-функцію m=h(M), після чого за допомогою відкритого ключа відправника перевіряє, чи відповідає отриманий підпис обчисленому значенню m геш-функції. Потім дешифрує своїм закритим ключем стегоключ, виймає з файлу стеговставку та порівнює її з шаблоном. Якщо перевірка підпису та цифрових водяних знаків пройшла успішно, то документ легітимний.

Геш-функція призначена для стиснення підписуваного документа M до декількох десятків або сотень біт. Геш-функція h() приймає як аргумент документ M довільного розміру і повертає геш-значення h(M)=H фіксованої довжини. Зазвичай геш-значення є стислим двійковим представленням основного

повідомлення довільної довжини. Слід зазначити, що значення геш-функції h(M) складним чином залежить від документа M і не дозволяє відновити сам документ M.

Геш-функція повинна задовольняти цілому ряду умов:

– геш-функція повинна бути чутлива до всіляких змін в тексті M, таких як вставки, викиди, перестановки і т.п.;

– геш-функція повинна мати властивість безповоротності, тобто підбір документу M', який би мав необхідне значення геш-функції, повинний бути неможливим;

– вірогідність того, що значення геш-функцій двох різних документів (незалежно від їх довжин) співпадуть, повинна бути нікчемно малою.

Тобто геш-функцією є таке математичне або алгоритмічне перетворення заданого блоку даних, яке володіє наступними властивостями:

– нескінченна область визначення;

– кінцева область значень;

– необоротність;

– зміна вхідного потоку інформації на 1 біт змінює близько половини всіх біт вихідного потоку.

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

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходе взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає

уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

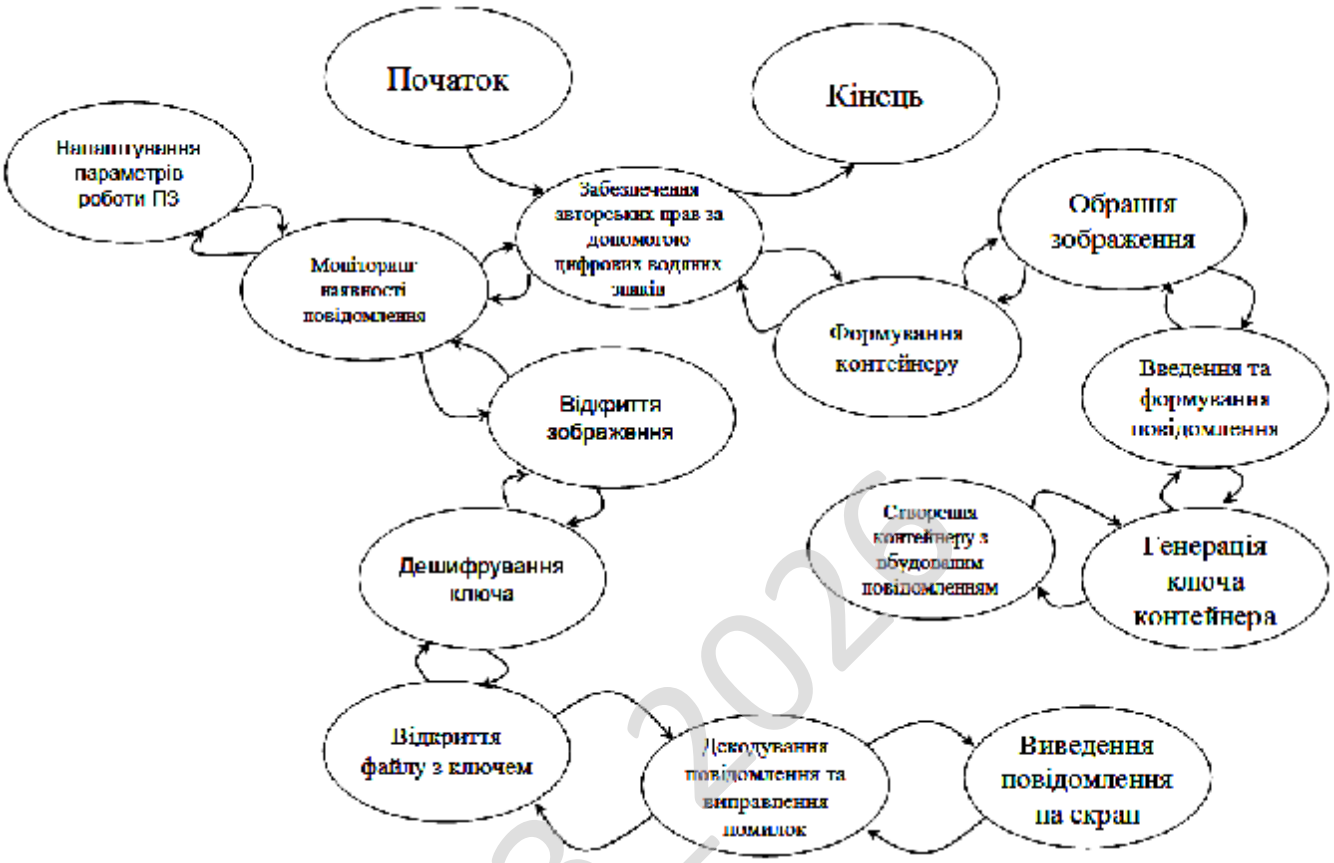


Рисунок 3.3 – Діаграма взаємодії процесів

Діаграми потоків даних містять чотири типи елементів:

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

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми. З якої видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

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

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об’єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків. При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

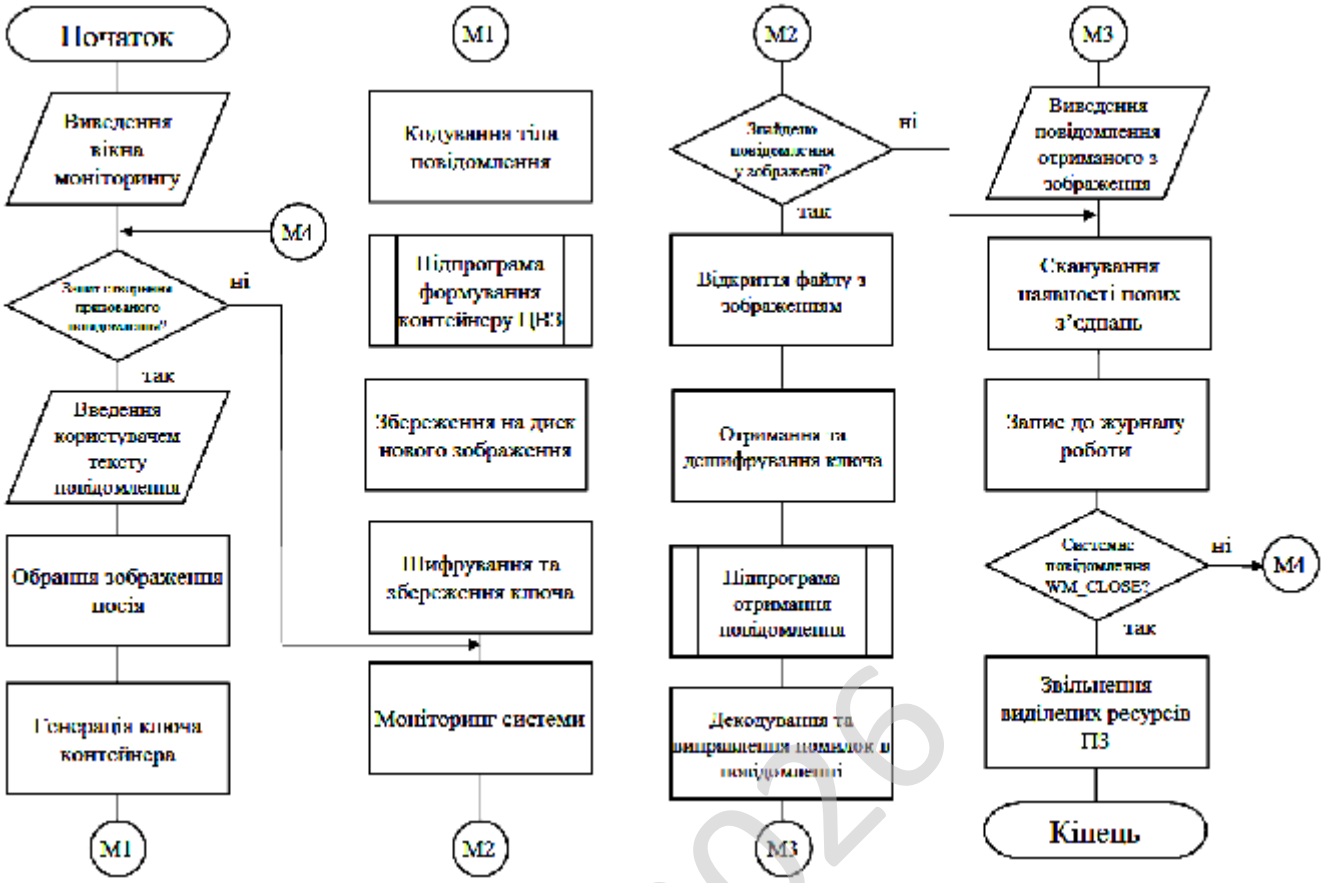


Рисунок 4.1 – Блок-схема основної програми

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. UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір

можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

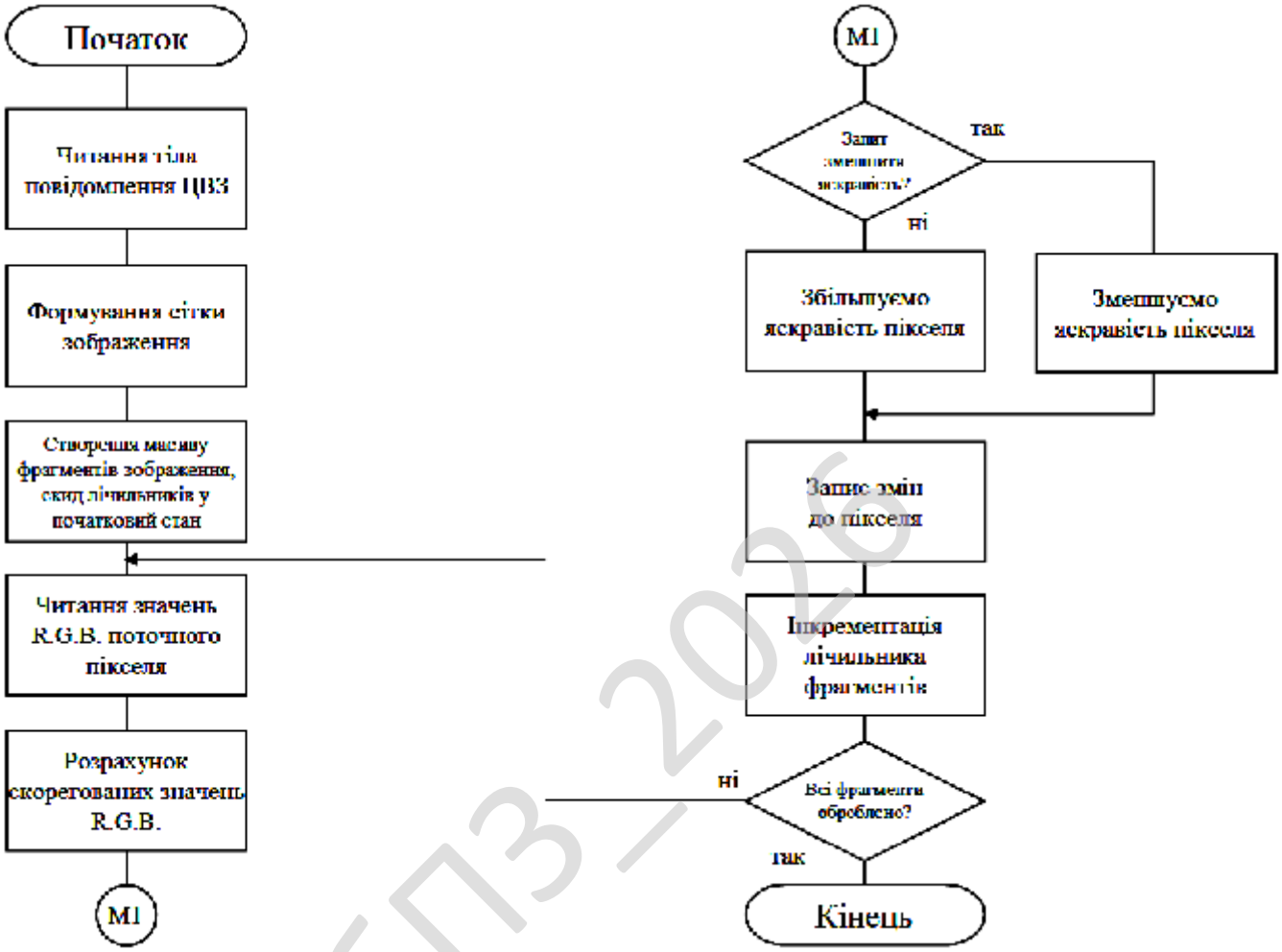


Рисунок 4.2 – Блок-схема роботи підпрограми

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об’єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

– Керівникам проектів, які керують розподілом завдань і контролем за проектом.

– Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.

– Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.

– Програмістам які реалізовують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

Також при розробці бакалаврської роботи було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів.

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.

Діаграма класів це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення.

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого

програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

В UML існують наступні типи зв'язків які використовуються у діаграмі класів: Асоціації; Агрегація; Композиція.

Асоціації це якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. Цілком припустимі випадки, коли обидва кінці асоціації відносяться до одного і того ж класу. Це означає, що з об'єктом деякого класу дозволено зв'язати інші об'єкти з того ж класу. Асоціація, що зв'язує два класи, називається бінарної. Можна, хоча це рідко буває необхідним, створювати асоціації, що зв'язують відразу кілька класів. Графічно асоціація зображується у вигляді лінії, що з'єднує клас сам з собою або з іншими класами.

Асоціації може бути присвоєно ім'я, яке описує природу відносини. Зазвичай ім'я асоціації не вказується, якщо тільки ви не хочете явно задати для неї рольові імена або у вашій моделі настільки багато асоціацій, що виникає необхідність посилатися на них і відрізняти один від одного. Ім'я буде особливо корисним, якщо між одними і тими ж класами існує кілька різних асоціацій.

Клас, що бере участь в асоціації, грає в ній деяку роль. По суті, це "обличчя", яким клас, що знаходиться на одній стороні асоціації, звернений до класу з іншого її боку. Можна явно позначити роль, яку клас грає в асоціації.

Часто при моделюванні буває важливо вказати, скільки об'єктів може бути пов'язано допомогою одного примірника асоціації. Це число називається кратністю (Multiplicity) ролі асоціації та записується або як вираз, значенням якого є діапазон значень, або в явному вигляді.

Вказуючи кратність на одному кінці асоціації, ви тим самим говорите, що на цьому кінці саме стільки об'єктів повинно відповідати кожному об'єкту на протилежному кінці. Кратність можна задати рівною одиниці (1), можна вказати діапазон: "нуль або одиниця" (0..1), "багато" (0 .. *), "одиниця або більше" (1 .. *). Дозволяється також вказувати певне число (наприклад, 3). За допомогою списку

можна задати і більш складні кратності, наприклад 0. . 1, 3..4, 6 .. *, що означає "будь-яке число об'єктів, крім 2 і 5".

Агрегація це проста асоціація між двома класами відображає структурний відношення між рівноправними сутностями, коли обидва класу знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший. Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин).

Ставлення такого типу називають агрегацією; воно зараховане до відносин типу «має» (з урахуванням того, що об'єкт-ціле має кілька об'єктів-частин). Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації з незафарбованим ромбом з боку «цілого». Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція це більш суворий варіант агрегації. Відома також як агрегація за значенням.

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбовані ромбиком.

Опис роботи системи

Програмне забезпечення системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків працює як прикладна Python система для захисту цифрових зображень.

Система вбудовує у файл прихований ідентифікатор автора, назву роботи, дату реєстрації та контрольний геш.

Після цього програма перевіряє наявність водяного знака та підтверджує належність зображення конкретному автору. Система використовує цифровий водяний знак як приховану службову інформацію, що зберігається всередині зображення без помітної зміни його вигляду.

Інтелектуальна частина системи аналізує збіг витягнутого знака з еталонними даними та формує рішення про справжність файлу.

Основні функції системи

– Завантаження цифрового зображення для захисту.

– Формування авторського ідентифікатора на основі даних автора та твору.

– Вбудовування цифрового водяного знака у молодші біти пікселів.

– Збереження захищеної копії зображення.

– Витягування водяного знака із зображення.

– Порівняння отриманого знака з контрольним значенням.

– Формування висновку про підтвердження або порушення авторських прав.

Розроблені основні модулі системи

– Модуль роботи із зображеннями.

– Модуль генерації цифрового водяного знака.

– Модуль вбудовування водяного знака.

– Модуль перевірки авторства.

– Модуль формування звіту.

Система має модульну архітектуру. Користувацький рівень приймає дані автора та файл зображення. Прикладний рівень створює цифровий водяний знак, вбудовує його у зображення та виконує перевірку.

Рівень зберігання містить захищені файли, контрольні геші та результати перевірок. Такий підхід спрощує супровід програми та розширення функцій захисту авторських прав.

Розглянемо фрагмент вихідного коду

```
from PIL import Image
import hashlib
import json
from datetime import datetime
class CopyrightWatermarkSystem:
    def __init__(self, secret_key):
        self.secret_key = secret_key
    def create_watermark(self, author, work_title):
```

```
#Формується службовий запис для підтвердження авторства
        data = {
            "author": author,
            "work_title": work_title,
            "date": datetime.now().strftime("%Y-%m-%d"),
            "secret_key": self.secret_key
        }
        raw_text = json.dumps(data, ensure_ascii=False)
watermark_hash = hashlib.sha256(raw_text.encode("utf-8")).hexdigest()
        return watermark_hash
    def text_to_bits(self, text):
#Текст цифрового водяного знака перетворюється у послідовність бітів
        bits = []
        for symbol in text:
            binary_symbol = format(ord(symbol), "08b")
            for bit in binary_symbol:
                bits.append(int(bit))
        return bits
    def bits_to_text(self, bits):

#Послідовність бітів відновлюється у текстовий водяний знак
        symbols = []
        for i in range(0, len(bits), 8):
            byte = bits[i:i + 8]
            if len(byte) == 8:
                symbols.append(chr(int("".join(map(str, byte)), 2)))
        return "".join(symbols)
    def embed_watermark(self, input_path, output_path, watermark):
        image = Image.open(input_path).convert("RGB")
        pixels = image.load()
        width, height = image.size
        watermark_bits = self.text_to_bits(watermark)
        watermark_bits += [0] * 8
        bit_index = 0

#Водяний знак вбудовується у молодший біт синього каналу пікселя
        for y in range(height):
            for x in range(width):
                if bit_index >= len(watermark_bits):
                    image.save(output_path)
                    return True
                r, g, b = pixels[x, y]
```

```
                b = (b & 254) | watermark_bits[bit_index]
                pixels[x, y] = (r, g, b)
                bit_index += 1
raise ValueError("Зображення замале для вбудовування водяного знака")
    def extract_watermark(self, image_path, max_symbols=64):
        image = Image.open(image_path).convert("RGB")
        pixels = image.load()
        width, height = image.size
        bits = []
        max_bits = max_symbols * 8

#Із молодших бітів синього каналу витягується прихований знак
        for y in range(height):
            for x in range(width):
                if len(bits) >= max_bits:
                    return self.bits_to_text(bits)
                r, g, b = pixels[x, y]
                bits.append(b & 1)
        return self.bits_to_text(bits)
    def verify_author_rights(self, image_path, author, work_title):
        expected_watermark = self.create_watermark(author, work_title)
        extracted_watermark = self.extract_watermark(image_path)

#Система порівнює очікуваний знак із фактично знайденим у файлі
        if expected_watermark in extracted_watermark:
            return "Авторські права підтверджено"
        return "Авторські права не підтверджено"
def main():
    sys
tem = CopyrightWatermarkSystem("student_secret_key_2026")
    author = "Іван Петренко"
    work_title = "Навчальна ілюстрація для дипломної роботи"
    watermark = system.create_watermark(author, work_title)
    system.embed_watermark(
        input_path="original_image.png",
        output_path="protected_image.png",
        watermark=watermark
    )
    result = system.verify_author_rights(
        image_path="protected_image.png",
        author=author,
        work_title=work_title
```

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

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Sinople – симетричний блоковий криптоалгоритм, побудований на основі незбалансованої «мережі Фейстеля». Алгоритм розроблено у 2003 році.

Основні вимоги до алгоритму при його розробці:

– Можливість програмної і апаратної реалізації.

– Висока швидкість.

– Простота.

– Низькі вимоги до пам'яті.

– Високий рівень безпеки.

Алгоритм заснований на 32-розрядних операціях і має 64 раунду, серед яких два типи – C і D. D раунди спроектовані для досягнення максимальної дифузії, C раунди – для досягнення перемішування. F-функція D раунду використовує один з елементів блоку даних (D[3]) та поточного з'єднання (K[r]) для трансформації 3-х елементів блоку даних. F-функція C раунду, навпаки, використовує перші три елемента блоку даних і поточний з'єднання (K[r]) для трансформації останнього елемента блоку даних (D[3]). Раунди D-типу виконуються до раундів C-типу. Додавання ключів з даними проводиться тільки через таблиці замін. Операції XOR (додавання за модулем 2) обов'язково поєднуються з операціями ADD (додавання за модулем 2^{32}).

Таблиці замін спочатку запозичені з алгоритму MARS і містять 512 32-розрядних елементів, проте були жорстко проаналізовано на предмет посилення.

Ключове розклад було спроектовано з урахуванням вимог:

– Простота

– Використовується та ж процедура, що і при шифруванні та розшифрування

– Установка ключа займає менше часу, ніж зашифрування

– Виключення еквівалентних ключів

– Виключення слабких ключів

Алгоритм, згідно із заявою авторів, стійкий до лінійного і диференціального аналізу.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання бакалаврської роботи.

Розроблене програмне забезпечення системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків складається з наступних функціональних блоків:

– Навігаційне меню: Дані; Шаблони; Налаштування; Довідка.

– Підвікно виведення отриманого зображення, з можливістю виділення блоків маркування.

– Вікна обрання групи .

– Вікно виведення результату роботи системи.

– Функціональних кнопок ПЗ.

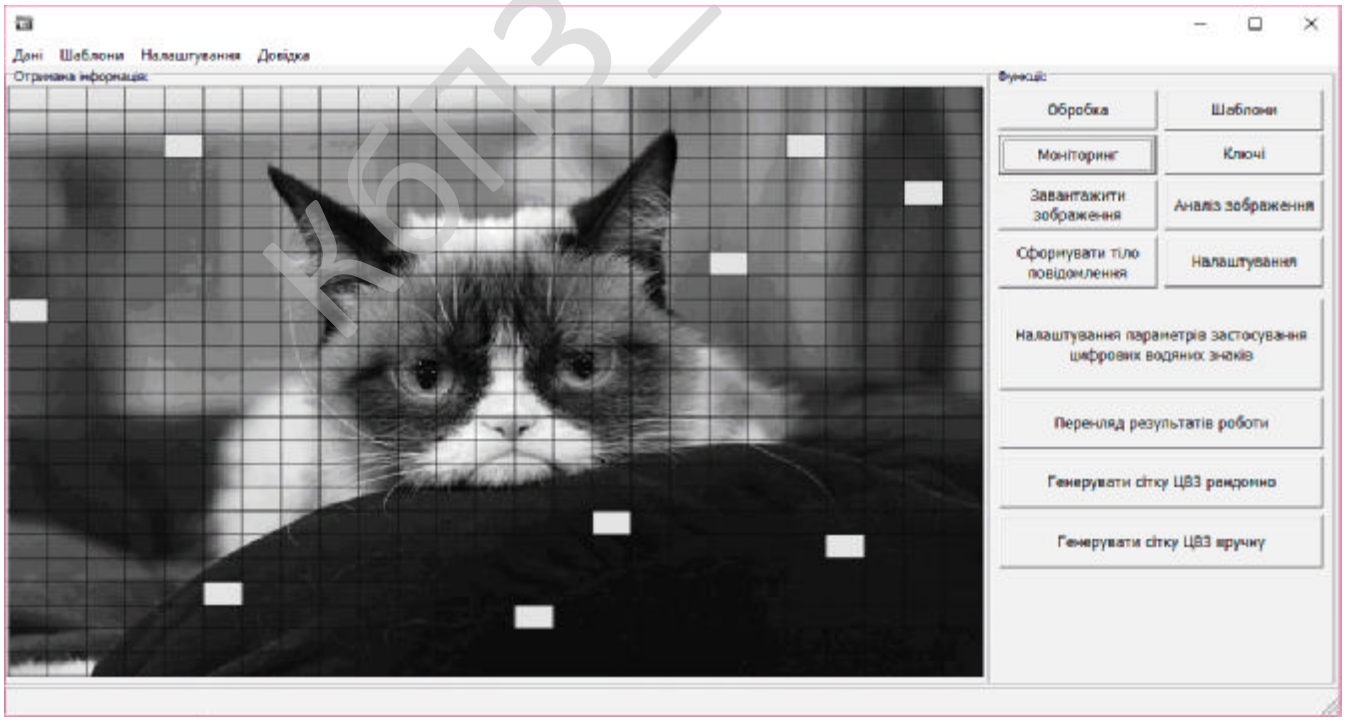


Рисунок 5.1 – Головне вікно розробленого ПЗ

Для перегляду короткої довідки про програму слід натиснути на основному вікні кнопку авторського права, після чого на екрані з’явиться вікно показане на рисунку 5.2.

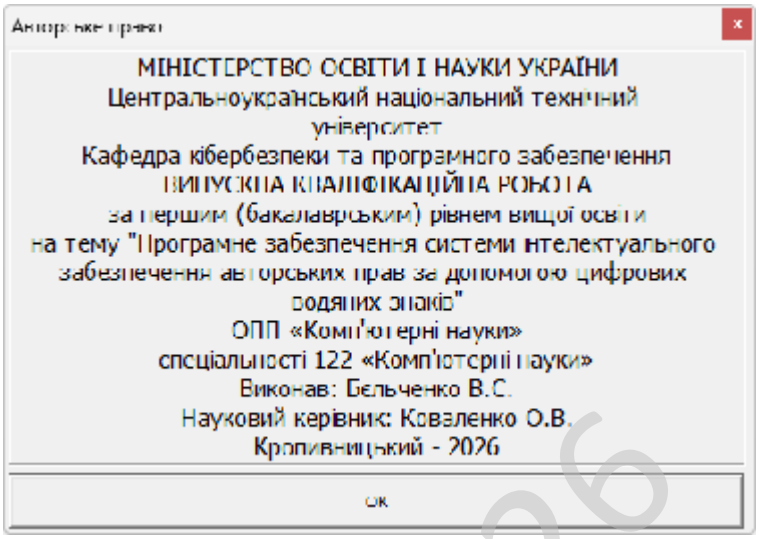


Рисунок 5.2 – Вікно розробника ПЗ

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.

Проводилось тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

– Гарантується перевірка всіх незалежних маршрутів програми.

– Знаходяться гілки True, False для всіх логічних рішень.

– Виконуються всі цикли (у межах їхніх кордонів та діапазонів).

– Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

– Кількість незалежних маршрутів може бути дуже велика.

– Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

– У програмі можуть бути пропущені деякі маршрути.

– Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

– Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

– Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

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 інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

– Досліджена система інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

– На основі отриманих результатів досліджень створена програмна реалізація системи інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків.

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 інтелектуального забезпечення авторських прав за допомогою цифрових водяних знаків. 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 Sinople.

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. 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.

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

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

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

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

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

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

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

9. 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.

10. 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.

11. 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.

12. Смірнова, Т.В. «Дослідження методів, моделей та сучасних IT-рішень для підтримки технологічних процесів у критичній інфраструктурі

держави». *Кібербезпека: освіта, наука, техніка.* 2025. Том 2 № 30. С.195-208, 2025.

13. 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.

14. 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.

15. 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.

16. 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.

17. 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.

18. Kuznetsov, O., Poluyanenko, N., Smirnov, O., Kozhakhmetova, D. «Optimized Simulated Annealing for Efficient Generation of Highly Nonlinear S-Boxes». *Advancements in Cybersecurity Next Generation Systems and Applications,* 2025. 146-174

19. Kuznetsov, O., Poluyanenko, N., Smirnov, O., Bekeshova, G. «Enhanced Cryptographic Security through Advanced S-Box Optimization: A Hybrid Heuristic Approach». *Advancements in Cybersecurity Next Generation Systems and Applications*, 2025. pp. 56-78.

20. Kuznetsov, O., Poluyanenko, N., Smirnov, O., Abduraimova, B «Enhancing Cryptographic Strength: A Novel Approach to S-Box Generation Using Modified Simulated Annealing». *Advancements in Cybersecurity Next Generation Systems and Applications*, 2025. pp. 79-116.

21. Kuznetsov, O., Derevianko, Y., Frontoni, E., Arnesano, M., Smirnov, O. «Factorial Representation of S-Boxes: A Novel Approach to Cryptographic Analysis and Optimization». *Advancements in Cybersecurity Next Generation Systems and Applications,* 2025. pp. 117-145.

22. Kuznetsov, O., Poluyanenko, N., Smirnov, O., Shaikhanova, A., Khruskov, B. «Innovative Cost Functions for Optimizing Cryptographic S-Box Generation». *Advancements in Cybersecurity Next Generation Systems and Applications*, 2025. pp. 29-55.

23. Kuznetsov, O., Smirnov, O., Akhmetov, B., Alimseitova, Z., Imoize, A.L. «Deep Learning Frontiers in Copy-Move Forgery Detection: Advances, Challenges, and Future Directions». *Advancements in Cybersecurity Next Generation Systems and Applications,* 2025. 202-229.

24. 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.

25. Kuznetsov, O., Smirnov, O., Mormul, M., Kotukh, Y., Zvieriev, V. «Comparative Research on Cryptocurrency Efficiency: An Objective Analysis of Key Metrics». *International Journal of Computing* 23(4), 2024. pp. 563-573.

26. Kuznetsov O., Frontoni E., Kuznetsova K., Smirnov O., Kostenko V. «Blockchain applications in metaverse environments: new horizons». *Advanced Metaverse Wireless Communication Systems.* pp. 255-293. 2024.

27. Kuznetsov, O., Frontoni, E., Chevardin, V., Smirnov, O., Imoize, A.L. «Advancing metaverse security with cryptographic innovations». *Advanced Metaverse Wireless Communication Systems.* pp 351-386. 2024.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. Kuznetsov, O., Frontoni, E., Kandiy, S., Smirnova, T., Prokopov, S., Bilanovych, A. «New Cost Function for S-boxes Generation by Simulated Annealing Algorithm». Lecture Notes on Data Engineering and Communications Technologies, 2023. vol 180. pp. 310-320. Springer, Cham.

35. Kuznetsov, O., Frontoni, E., Kandiy, S., Smirnov, O., Ulianovska, Y., Kobylianska, O. «Heuristic Search for Nonlinear Substitutions for Cryptographic Applications». *Lecture Notes on Data Engineering and Communications Technologies,* 2023. vol 180. Springer, Cham. pp. 288-298.

36. Kuznetsov, O., Kuznetsova, Y., Smirnov, O., Kostenko, O., Zvieriev, V. «Evaluating Hashing Algorithms in the Age of ASIC Resistance». *CEUR Workshop Proceedings*, 2023, 3628, pp. 93-105.

37. Kuznetsov O., Frontoni E., Kuznetsova Ye., Smirnov O., Chevardin V. «Achieving Enhanced Security in Biometric Authentication: A Rigorous Analysis of Code-Based Fuzzy Extractor». *CEUR Workshop Proceedings,* Volume 3624, 2023, pp. 330-339.

38. 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.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

44. 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.

45. 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,

46. 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.

47. 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.

48. 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.

49. 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.

50. 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.

51. Смірнов О.А., Конопліцька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Книшук А.В. «Вступ до кібербезпеки»: навчальний посібник – Кропивницький: ЦНТУ – 2022. – 968 с.

52. Теорія та практика сучасного інформаційно-психологічного протиборства: навчальний посібник / [В.М. Петрик, С.О. Гнатюк, М.М. Присяжнюк та ін.]; за заг. ред. С.О. Гнатюка, В.М. Петрика та О.А Смірнова. – Полтава, 2022. – 334 с.

53. 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

54. 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.

55. 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.

56. 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.*

57. 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.