Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки та захисту
інформації
_____ Іван ПАРХОМЕНКО
«13» червня 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)

спеціальність 125 Кібербезпека
(код і назва спеціальності)

освітній ступень бакалавр

освітня програма Кібербезпека
(назва освітньо-професійної програми)

на тему: «Стеганографічна система приховання інформації у недрукованих символах електронних документів»

Виконавець: студент IV курсу, групи КБ-42

_____ **Вадим ПАНЧЕНКО**
(підпис) (ім'я, прізвище)

	Підпис	**Ім'я ПРІЗВИЩЕ**
Керівник		Юрій БАБЕНКО
Нормоконтроль		Інна МИХАЛЬЧУК

Київ 2025

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«29» листопада 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньо-професійної програми)

Студенту **КБ-42** (група) **Панченку Вадиму Сергійовичу** (прізвище ім'я по батькові)

Тема кваліфікаційної роботи Стеганографічна система приховання інформації у недрукованих символах електронних документів

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №6 від 28.11.2024 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Структура та архітектура DOCX-документів, принципи прихованого додавання даних у текстове середовище.

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Основні концепції текстової стеганографії, методи вбудовування даних у текстові документи, реалізація програмного рішення.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Програмне рішення, яке спрямоване на приховане додавання та витяг текстової інформації з документів.

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 29 листопада 2024 року

Завдання видав _____ (підпис) Юрій БАБЕНКО (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Вадим ПАНЧЕНКО (ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт	Відмітка про виконання
1	Уточнення постановки задачі	29.11.2024 – 22.01.2025	*виконано*
2	Аналіз літератури	23.01.2025 – 11.02.2025	*виконано*
3	Обґрунтування вибору рішення	12.02.2025 – 15.02.2025	*виконано*
4	Дослідження принципів і класифікацій методів приховання інформації в цифрових документах.	16.02.2025 – 04.03.2025	*виконано*
5	Огляд архітектури та особливостей форматів DOC і DOCX як середовищ для стеганографії.	05.03.2025 – 21.03.2025	*виконано*
6	Визначення вразливостей та потенційних точок вбудовування даних у структуру Word-документів	22.03.2025 – 08.04.2025	*виконано*
7	Реалізація програмного засобу приховання та зчитування повідомлень із DOCX-файлів.	09.04.2025 – 10.05.2025	*виконано*
8	Оформлення пояснювальної записки	11.05.2025 – 27.05.2025	*виконано*
9	Підготовка до захисту кваліфікаційної роботи	28.05.2025 – 13.06.2025	*виконано*

Завдання видав _____ (підпис) Юрій БАБЕНКО (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Вадим ПАНЧЕНКО (ім'я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 13 червня 2025 року

РЕФЕРАТ

Кваліфікаційна робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, додатків, має 57 сторінок основного тексту, 5 таблиць та 4 рисунки. Список використаних джерел містить 30 найменуваннь і займає 4 сторінки.

Метою кваліфікаційної роботи є створення системи стеганографічного приховування даних у недрукованих символах електронних документів.

Об'єктом дослідження є процес приховування інформації у недрукованих символах електронних документів.

Предметом дослідження є сукупність методів, прийомів і стратегій стеганографічного приховування даних у недрукованих символах електронних документів та сумісності з типовими засобами обробки текстових файлів.

Для досягнення зазначеної мети були поставлені наступні завдання:

- проаналізувати теоретичні основи стеганографії як методу прихованого впровадження інформації;
- провести аналіз структурних та функціональних особливостей документів MS Word як потенційного носія стеганографічної інформації;
- розробити програмний засіб для приховання інформації у недрукованих символах Word-документів, реалізувавши алгоритми кодування та вилучення.

Практична цінність одержаних результатів полягає у розробці та впровадженні програмного засобу, що дозволяє здійснювати приховане впровадження інформації в документи формату MS Word шляхом модифікації недрукованих символів без помітного впливу на візуальну структуру та форматування тексту.

Ключові слова: текстова стеганографія, приховування інформації, недруковані символи, інформаційна безпека, програмне забезпечення.

ЗМІСТ

ВСТУП

Розвиток цифрових технологій та ускладнення архітектури електронного документообігу відкрили нові горизонти не лише для зручності зберігання й передачі інформації, а й для застосування нетрадиційних методів її захисту. Однією з таких технологій є стеганографія — наука про приховування інформації в межах інших, на перший погляд, звичайних цифрових об’єктів. Особливої актуальності набуває застосування стеганографічних методів у текстових документах, де візуальні зміни можуть бути мінімальними або повністю непомітними.

Найвразливішими до прихованих вставок виявляються електронні документи формату DOCX, що базуються на відкритій XML-структурі та містять численні службові області: від стилів і форматів до метаданих і прихованих символів. Ці особливості створюють можливості не лише для прихованого обміну конфіденційною інформацією, але й для потенційного зловживання — зокрема, вбудовування шкідливих інструкцій або прихованих ідентифікаторів, які важко виявити звичайними засобами.

У практиці інформаційної безпеки дедалі частіше фіксуються випадки використання недрукованих символів Unicode, таких як Zero Width Space (U+200B) або Zero Width Joiner (U+200D), для приховування повідомлень у текстах. Такі символи не впливають на зовнішній вигляд документа, але фізично присутні в його структурі, що дозволяє використовувати їх як носії інформації у стеганографічних системах. Актуальність теми кваліфікаційної роботи обумовлена необхідністю розробки інструментів, які дозволяють надійно приховувати дані в текстових документах без порушення їх семантичної та візуальної цілісності.

Метою даної кваліфікаційної роботи є створення системи стеганографічного приховування даних у недрукованих символах документів MS Word.

Для досягнення зазначеної мети були поставлені наступні завдання:

- проаналізувати теоретичні основи стеганографії як методу прихованого впровадження інформації;
- провести аналіз структурних та функціональних особливостей документів MS Word як потенційного носія стеганографічної інформації;
- розробити програмний засіб для приховання інформації у недрукованих символах Word-документів, реалізувавши алгоритми кодування та вилучення.

Об’єктом дослідження є процес приховування інформації у текстових документах MS Word.

Предметом дослідження є сукупність методів, прийомів і стратегій стеганографічного приховування даних у недрукованих символах документів MS Word та сумісності з типовими засобами обробки текстових файлів.

Практичне значення кваліфікаційної роботи полягає у створенні програмного засобу, здатного приховувати текстову інформацію в електронних документах формату DOCX за допомогою недрукованих символів, що не змінюють візуального сприйняття документа для кінцевого користувача. Розроблена система може бути використана в інформаційних середовищах, де необхідно передавати конфіденційні дані без виявлення факту їх існування, зокрема — в умовах підвищеного контролю, у внутрішньокорпоративному обміні або при захисті авторських прав. Крім того, розроблене рішення може слугувати основою для подальших досліджень у сфері текстової стеганографії, зокрема для створення комбінованих систем захисту, які поєднують методи приховання з криптографічними засобами або засобами автентифікації.

Галузь застосування розробленої стеганографічної системи охоплює сфери, пов’язані з кібербезпекою організацій, що працюють з електронною документацією та потребують прихованих механізмів передачі інформації без виявлення її наявності. Насамперед це державні установи, науково-дослідні центри, фінансові та юридичні організації, підприємства критичної інфраструктури, а також структури, що забезпечують інформаційний супровід

стратегічних процесів. Застосування методів приховання у недрукованих символах дозволяє непомітно вбудовувати службові повідомлення, ідентифікаційні маркери, контрольні підписи або інші елементи захищеної комунікації в звичайні офісні документи, що істотно знижує ймовірність їх перехоплення, модифікації або витоку. Система може ефективно використовуватись для маркування документів із метою виявлення витоків, а також для прихованого обміну даними в умовах публічного або контрольованого цифрового середовища.

Наукова новизна розробленої роботи полягає у створенні прикладної стеганографічної системи, яка реалізує метод приховання інформації в електронних документах формату DOCX за допомогою недрукованих символів Unicode у поєднанні з модифікацією кольорових атрибутів текстових фрагментів. Запропонований підхід відрізняється від класичних методів текстової стеганографії тим, що не порушує візуальну структуру документа та забезпечує високу стійкість до базових форматних змін, таких як редагування, збереження чи копіювання. Крім того, система забезпечує автономну декодування прихованих повідомлень без необхідності попереднього форматування або ручного аналізу, що підвищує її практичну придатність у реальних умовах використання. Розроблений інструмент передбачає гнучку стратегію розміщення вставок (лінійну, псевдовипадкову або структурну), що дозволяє адаптувати приховання під конкретні умови або рівень ризику виявлення.

Практична цінність одержаних результатів полягає у розробці та впровадженні програмного засобу, що дозволяє здійснювати приховане впровадження інформації в документи формату MS Word шляхом модифікації недрукованих символів без помітного впливу на візуальну структуру та форматування тексту.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ СТЕГАНОГРАФІЇ ЯК ЗАСОБУ ПРИХОВАНОГО ВПРОВАДЖЕННЯ ІНФОРМАЦІЇ

1.1 Поняття прихованого впровадження даних у цифровому середовищі

В період цифровізації, коли більшість комунікацій та операцій із даними відбувається в електронному форматі, питання прихованого передавання інформації набуває не лише прикладного, а й стратегічного значення. Окрім традиційних засобів захисту, таких як криптографічне шифрування, автентифікація чи контроль доступу, все ширше застосовуються методи, що дозволяють забезпечити приховану присутність даних у цифровому середовищі без очевидного виявлення їх сторонніми спостерігачами [1].

Приховане впровадження даних передбачає включення додаткової, зазвичай конфіденційної або службової, інформації в основний цифровий носій таким чином, щоб зберегти його функціональні та візуальні властивості незмінними. У цьому контексті особливого значення набувають ті способи приховування, що не порушують структури або вмісту даних на рівні, який міг би бути легко помічений під час звичайного перегляду чи автоматичної обробки. Саме властивість приховати сам факт передавання повідомлення відрізняє стеганографічні методи від класичної криптографії, яка, хоча й шифрує зміст, залишає очевидним сам факт комунікації.

У широкому сенсі поняття інформаційного приховування охоплює кілька близьких за метою підходів. Зокрема, стеганографія спрямована на маскування самої наявності повідомлення шляхом його вбудовування в інший, нейтральний або неінформативний носій. Цифрове водяне маркування, хоч і використовує схожі технічні принципи, має дещо іншу мету — забезпечити ідентифікацію прав власності, відстеження джерел розповсюдження або контроль цілісності

цифрового контенту. Іншим напрямом є створення ковертних (прихованих) каналів передачі даних, які дозволяють здійснювати комунікацію в обхід стандартних протоколів моніторингу чи безпеки [2].

Особливої уваги в межах стеганографії заслуговує робота з текстовими носіями, які, з одного боку, є найбільш поширеною формою представлення інформації, а з іншого — мають низький рівень надлишковості та високу чутливість до змін. Це суттєво ускладнює завдання приховування: будь-яка зміна структури або форматування може призвести до втрати або спотворення вбудованих даних.

1.2 Загальні принципи стеганографічних методів

Стеганографія, як окрема галузь у системі інформаційної безпеки, зосереджується не стільки на захисті вмісту передаваних даних від несанкціонованого прочитання, скільки на забезпеченні їх повної невидимості в каналі комунікації. Її ключова функція полягає у приховуванні самого факту існування повідомлення, що радикально відрізняє цей підхід від криптографії, основним завданням якої є трансформація відкритого тексту у форму, непридатну для інтерпретації сторонніми особами.

Основа стеганографії передбачає інтеграцію прихованої інформації в цифровий об’єкт, який вже має інформаційне наповнення. Таким чином, носій виконує подвійну роль — як засіб звичайного обміну даними і водночас як канал для непомітного вбудовування додаткових відомостей. З технічної точки зору це реалізується шляхом модифікації слабко контрольованих або надлишкових елементів структури носія: найменш значущих бітів, пробілів, міжрядкових інтервалів, значень кольору тощо. При цьому критично важливо, щоб такі модифікації не порушували функціональність документа, не викликали підозри при візуальному перегляді та залишались стійкими до стандартних процедур обробки — редагування, конвертації або пересилання.

Цей підхід принципово відрізняється від криптографії не лише за метою, а й за логікою взаємодії з носієм. Якщо криптографічні алгоритми створюють явний, хоча й зашифрований інформаційний об’єкт (наприклад, файл з розширенням .enc або зашифрований текст, що має хаотичну структуру), то стеганографія оперує в межах звичних форматів — зображень, текстових файлів, аудіо- та відеоматеріалів — не залишаючи явних слідів втручання. Така «невидимість» є вирішальною перевагою в умовах, коли навіть сам факт застосування шифрування може бути підставою для підозри або обмеження доступу [3].

Більше того, стеганографічні методи можуть використовуватись у поєднанні з криптографією: спочатку повідомлення шифрується, а вже потім зашифрований вміст вбудовується в носій. Це дозволяє не лише приховати повідомлення, а й забезпечити його конфіденційність у випадку виявлення.

1.2.2 Класифікація стеганографічних методів за типом носія

Класифікація стеганографічних методів здійснюється на основі низки параметрів, серед яких основоположним є тип цифрового носія, що використовується для приховання інформації.

Відповідно до цього критерію, виділяють графічну, аудіо-, відео-, мережеву та текстову стеганографію [4]. Ознайомитись з характеристиками типів носіїв можна у табл. 1.1.

У графічних методах приховані дані вбудовуються в пікселі зображення, зазвичай шляхом модифікації найменш значущих бітів. Такі методи є одними з найпоширеніших завдяки високій надлишковості зображень, яка дозволяє зберігати значні обсяги інформації без візуального викривлення.

Аудіо- та відеостеганографія працюють із сигналами у часовій або частотній області, змінюючи спектральні або амплітудні характеристики. Хоча такі підходи вимагають складніших алгоритмів обробки, вони забезпечують добру місткість і високу стійкість до звичайних перетворень.

Мережева стеганографія, у свою чергу, реалізується через зміну параметрів передавання даних у комп’ютерних мережах. Приховані повідомлення можуть бути закладені у затримки між пакетами, черговість їх надсилання або варіативність у полях заголовків. Цей напрям особливо актуальний для організації прихованих каналів зв’язку в умовах суворого контролю.

Серед усіх типів носіїв найбільш чутливим і технічно складним для стеганографічного впровадження є текст. Через низьку надлишковість і високу структурну щільність, будь-які зміни у текстовому документі легко можуть бути помічені як людиною, так і автоматизованими засобами.

Таблиця 1.1

Типи носіїв та їх характеристика

Тип носія	**Характеристика**
Графічні	Вбудовування даних у пікселі зображень
Аудіо	Модифікація амплітуди або частот звукового сигналу
Відео	Використання окремих кадрів або потоків даних
Мережеві	Передача даних через протоколи, пакети, час затримки
Текстові	Вбудовування інформації у структуру або форматування тексту
Метадані	Приховування інформації у службовій частині файлів

1.2.3 Активні та пасивні методи впровадження

Крім класифікації за типом носія, не менш важливим є також спосіб впровадження інформації, що визначає характер взаємодії з цифровим об’єктом. Залежно від цього критерію, методи стеганографії поділяються на активні та пасивні. Цей поділ є принциповим з погляду впливу на цілісність носія, стійкість до виявлення та рівень технічної складності реалізації.

Активні методи передбачають внесення явних, хоча й мінімальних, змін до структури цифрового носія. Це можуть бути, зокрема, модифікації бітів у графічному файлі, вставка додаткових пробілів або табуляцій у текстовому документі, зміна тривалості аудіосигналу або корекція значень у заголовках мережевих пакетів. Головною перевагою активного підходу є можливість точного контролю місця й способу розміщення інформації, що дозволяє забезпечити вищу місткість. Проте така модифікація несе ризик залишення цифрових слідів, які можуть бути виявлені під час стеганалітичного аналізу, особливо в умовах повторної обробки або стискання носія [5].

Пасивні методи не змінюють сам носій, а використовують його вже наявні особливості для приховання інформації. У текстових документах це може бути, наприклад, черговість абзаців, розміщення стилістичних блоків, варіативність синонімів чи порядок заголовків. У мережевих протоколах — це часові інтервали між переданими пакетами або шаблони запитів. Пасивний підхід має значно вищу стійкість до виявлення, оскільки не створює помітних змін на рівні структури або візуального сприйняття. Проте він обмежений у плані місткості та залежить від контексту: приховати великий обсяг інформації без прямого втручання вміст носія зазвичай неможливо.

1.3 Основні характеристики та вимоги до стеганографічних систем

Оцінювання ефективності стеганографічних методів неможливе без урахування базових характеристик, які визначають здатність системи

забезпечувати приховану передачу інформації в реальних умовах функціонування цифрових носіїв. Ідеться не лише про сам факт успішного вбудовування повідомлення, а передусім про збереження його непомітності, цілісності й здатності до відновлення після можливих перетворень. Виділяють три ключові параметри, а саме: непомітність**,** місткість і стійкість, кожен з яких виконує свою роль у загальній надійності стеганографічної системи. Зважаючи на взаємозалежність цих характеристик, побудова ефективного методу приховування завжди передбачає пошук компромісу між їхнім оптимальним співвідношенням.

Непомітність визначає ступінь збереження зовнішнього вигляду або поведінки цифрового носія після вбудовування повідомлення. Інакше кажучи, це характеристика, яка вказує на те, наскільки внесення прихованої інформації не змінює основного контенту з точки зору спостерігача — людини або автоматизованої системи. У разі текстових документів мова йде, зокрема, про відсутність змін у розташуванні символів, міжрядкових інтервалах, стилях, які могли б бути виявлені під час звичайного перегляду або внаслідок порівняльного аналізу. Високий рівень непомітності є необхідною умовою для приховування самого факту комунікації, особливо в умовах суворого контролю за інформаційним простором.

Місткість стосується максимальної кількості інформації, яку можливо вбудувати в цифровий носій без порушення його структури та без втрати непомітності. Це критично важливий параметр для практичних систем, де потрібно передати не лише сигнальний біт, а достатній обсяг смислового або службового повідомлення. У випадку текстових документів, зокрема у форматі MS Word, місткість обмежується особливостями форматування, наявністю службових тегів, структурою XML-компонентів та іншими аспектами, які можуть бути використані без візуального викривлення документа. Низька надлишковість таких носіїв вимагає високої точності під час кодування, а також розробки нестандартних методів, орієнтованих на максимальне використання технічних особливостей формату.

Стійкість відображає здатність прихованої інформації зберігатися у випадку трансформації, обробки або часткового пошкодження носія. Йдеться про такі операції, як повторне збереження документа, конвертація між форматами, автоматичне очищення форматування, копіювання або пересилання через зовнішні сервіси. У деяких випадках, особливо при використанні активних методів, навіть незначна зміна структури може призвести до повної втрати вбудованих даних або до їх пошкодження. Тому стійкість часто вимагає вбудовування резервних кодів, надлишкового кодування, або реалізації самовідновлюваних схем, що підвищує загальну складність системи [6,7].

Жоден із наведених критеріїв не є домінуючим у всіх сценаріях застосування. Залежно від цілей пріоритет може надаватися тій чи іншій характеристиці. У практичних реалізаціях часто доводиться обирати компроміс: збільшення місткості зазвичай знижує рівень непомітності, а посилення стійкості призводить до ускладнення структури повідомлення. Ознайомитись з характеристиками та їх описом можна у табл. 1.2.

Таблиця 1.2

Характеристики, які визначають здатність системи забезпечувати приховану передачу інформації

Характеристика	**Опис**
Непомітність	Здатність прихованого повідомлення залишатися непомічений для користувача або системи аналізу
Місткість	Максимальний обсяг даних, що може бути вбудований у носій без його помітного спотворення
Стійкість	Стійкість прихованих даних до змін, перетворень або пошкодження носія під час обробки

1.4 Сучасні напрями розвитку текстової стеганографії

Ні для кого не секрет, що текстові документи залишаються важливою, хоча й технічно складною сферою прихованого впровадження інформації. Їх значення зумовлене не лише поширеністю у повсякденному електронному спілкуванні, а й тим, що вони є основною формою представлення даних у таких сферах, як діловодство, юриспруденція, освіта та міжособистісна комунікація. Текстові носії — особливо у форматах MS Word, PDF чи HTML — слугують універсальним каналом передачі інформації, що робить їх привабливим середовищем для прихованих повідомлень, зокрема в умовах, коли використання шифрування може викликати підозру або бути обмеженим.

Проте на відміну від мультимедійних форматів, які характеризуються високою надлишковістю та здатністю маскувати інформацію у фоні (наприклад, у шумі зображення або спектрі звуку), текстові документи мають обмежену структурну гнучкість. Будь-яка незначна зміна — додатковий пробіл, зсув пунктуації, варіація формулювання — може бути помічена читачем або знищена під час редагування. Така висока чутливість до модифікацій робить процес приховування у тексті вразливим і вимагає застосування особливо тонких, лінгвістично обґрунтованих або форматно-специфічних методів.

Водночас саме широке використання текстових документів у відкритому цифровому обміні — через електронну пошту, корпоративні платформи, онлайн-редактори — створює велику кількість потенційних каналів для непомітного передавання інформації [8].

1.4.1 Тренди в дослідженнях стеганографії

Дослідження в галузі текстової стеганографії демонструють поступовий перехід від примітивних технік до складніших, багаторівневих моделей, орієнтованих на глибше використання внутрішньої структури документа та лінгвістичних закономірностей. Якщо раніше переважали прості прийоми на

кшталт вставки додаткових пробілів, табуляцій або непомітних символів, то нині спостерігається зміщення фокусу на семантично вмотивовані перетворення тексту, а також на роботу з метаданими, структурними атрибутами і службовими тегами складних офісних форматів.

Одним із провідних напрямів є використання недрукованих Unicode-символів, зокрема символів нульової ширини (zero-width space, zero-width joiner, zero-width non-joiner), що дозволяють кодувати інформацію без жодного візуального впливу на документ. Їх можна вставляти між символами або словами, не порушуючи при цьому зовнішнього вигляду тексту. Цей підхід набув особливої популярності у форматах, які підтримують збереження розширених символьних репрезентацій — наприклад, DOCX, де символна послідовність фіксується на рівні XML-представлення незалежно від візуального рендерингу.

Паралельно розвивається напрям лінгвістично орієнтованої стеганографії, що передбачає кодування інформації через зміни на рівні синтаксису або стилістики. До таких методів належать синонімічна заміна, перестановка слів, маніпуляції з порядком речень або варіації пунктуаційних структур. Високий рівень природності таких перетворень дозволяє зберігати непомітність, однак суттєво ускладнює реалізацію, оскільки потребує глибокого контекстуального аналізу й обережної роботи з мовними одиницями. Останні розробки у цьому напрямі передбачають використання засобів машинного навчання та мовних моделей, здатних адаптивно обирати місця і способи впровадження без втрати смислової цілісності.

Ще одним значущим трендом є робота з форматно-структурними властивостями документів, зокрема — використання тегів, властивостей стилів, коментарів, полів змісту, гіперпосилань, а також інших XML-елементів, що не завжди виводяться на екран. У форматі DOCX, наприклад, можливе приховування інформації в окремих підкомпонентах документу — settings.xml, styles.xml, або document.xml, — які містять технічні параметри, що не відображаються безпосередньо користувачеві, але зберігаються під час передачі та зчитуються при відкритті документа.

Сукупно ці тенденції можуть свідчити про перехід текстової стеганографії до фази складних, багатошарових моделей, які поєднують лінгвістичний аналіз, структурну оптимізацію та технічну гнучкість обраного формату. Водночас активізується розробка автоматизованих інструментів, які інтегрують у собі кілька стеганографічних підходів, даючи змогу адаптувати метод до конкретного типу документа, мови або контексту використання [9,10].

1.4.2 Виклики та обмеження у текстовому середовищі

Незважаючи на зростаючу складність і розмаїття методів, застосування стеганографії у текстових документах залишається технічно проблемним напрямом через низку об’єктивних обмежень, зумовлених особливостями самого текстового носія. На відміну від мультимедійних форматів, де надлишкова інформація дозволяє впроваджувати зміни без порушення загального сприйняття, текст характеризується високою щільністю змісту при мінімальному рівні надлишковості, що суттєво ускладнює непомітне приховування даних.

Насамперед, текстова інформація є візуально чутливою до змін. Навіть незначні модифікації, як-от додатковий пробіл, порушення вирівнювання або варіація пунктуації, можуть бути легко помічені користувачем. Особливо це актуально у випадках, коли документ має строго визначений шаблон або структуру (наприклад, юридичні чи звітні документи), де будь-яке відхилення викликає підозру. Відсутність природного «фону» або шумової області, у яку можна було б замаскувати дані, суттєво обмежує місткість і стійкість текстових методів.

Другою групою викликів є процеси автоматизованої обробки текстів. Переважна більшість текстових документів на практиці проходить через численні етапи редагування, збереження у нових форматах, стиснення або передавання через зовнішні платформи (наприклад, Google Docs, поштові сервери, системи керування документами). Під час таких операцій часто

відбувається автоматичне нормалізування форматування: видаляються зайві символи, оновлюються стилі, переформатовуються абзаци, що призводить до втрати або спотворення прихованої інформації. Це особливо стосується недрукованих символів або прихованих елементів структури, які можуть бути некоректно оброблені системою.

Крім того, лінгвістично орієнтовані методи хоч і забезпечують високий рівень непомітності, однак характеризуються низькою місткістю та високою залежністю від контексту. Наприклад, синонімічні заміни або перестановка компонентів речення потребують достатнього текстового обсягу та високого рівня лексичної варіативності, що не завжди доступно. Окрім цього, такі методи є мовно специфічними та не завжди масштабуються між різними мовами або жанрами документів [11].

Також слід ураховувати соціальні та організаційні аспекти. У корпоративному або офіційному середовищі документи часто проходять перевірку на відповідність шаблонам або наявність прихованих елементів (наприклад, вбудованих макросів, змінених стилів, незвичних властивостей метаданих). Сучасні системи інформаційної безпеки все частіше включають модулі базового стеганалізу, що знижує ефективність простих або повторюваних схем приховування. Ознайомитись більш детально з обмеженнями у застосуванні стеганографії, а саме з категоріями обмежень, суттю проблеми та наслідками для стеганографії можна у табл. 1.3.

Таблиця 1.3

Обмеження у застосуванні стеганографії

Категорія обмежень	**Суть проблеми**	**Наслідки для стеганографії**
Візуальна чутливість тексту	Навіть незначні зміни (пробіли, відступи, пунктуація) помітні для користувача	Обмеження методів приховування, особливо в стандартизованих або формалізованих документах
Відсутність «фону» або шуму	Текст не містить ділянок з природною варіативністю	Обмежена місткість та знижена стійкість прихованої інформації
Автоматизована обробка тексту	Нормалізація форматування, стиснення, конвертація під час редагування або пересилання	Високий ризик втрати чи пошкодження прихованих даних, особливо недрукованих або структурних символів
Лінгвістичні обмеження	Залежність від контексту, обсягу тексту, мовної специфіки, складність синтаксичних змін	Низька місткість; складність масштабування на різні мови або жанри
Організаційно-соціальні чинники	Впроваджені перевірки шаблонів, контроль метаданих, використання систем стеганалізу	Підвищений ризик виявлення прихованої інформації; необхідність розробки більш складних і нестандартних схем

1.5 Відмінність текстової стеганографії від криптографії та цифрових водяних знаків

В актуальних підходах до захисту інформації стеганографія займає специфічне місце, оскільки на відміну від криптографії чи цифрового маркування, її мета полягає не стільки у захисті вмісту, скільки у приховуванні самого факту існування інформації. Водночас через спільну сферу застосування — а саме забезпечення конфіденційності, автентичності та контролю доступу до даних — ці поняття нерідко плутають або ототожнюють. Тому важливо чітко визначити їх функціональні відмінності, призначення та характер впливу на цифрові об’єкти.

Криптографія — це науково-прикладна дисципліна, що зосереджена на перетворенні відкритої інформації у форму, непридатну для інтерпретації без спеціального ключа. В результаті шифрування одержується так званий шифротекст, який, хоча і не розкриває змісту, чітко сигналізує про наявність зашифрованих даних. Це означає, що криптографічне перетворення приховує лише зміст повідомлення, але не сам факт його існування. У випадках, коли використання шифрування є підставою для посиленого контролю або викликає підозру, криптографія не здатна забезпечити повну інформаційну невидимість[12].

Стеганографія, на відміну від цього, має на меті маскування самого факту передачі даних. Повідомлення вбудовується у цифровий об’єкт таким чином, щоб бути непомітним і не виявляти своєї присутності навіть під час перегляду або первинної технічної обробки. Вона не забезпечує шифрування вмісту (хоча часто поєднується з криптографією для підвищення безпеки), але натомість гарантує прихованість комунікаційного акту. Найчастіше використовується у випадках, коли потрібен повний «інформаційний камуфляж» у публічних або підконтрольних середовищах.

Цифрові водяні знаки — це ще один тип інформаційного впровадження, який, переслідує зовсім інші цілі. Головне його призначення — захист

авторських прав, ідентифікація джерела або контроль за розповсюдженням цифрового контенту. При цьому цифровий водяний знак, на відміну від стеганографії, зазвичай є стійким до перетворень (наприклад, перекодування, стиснення, обрізання) і може зберігатися у вмісті навіть після багаторазового копіювання. У деяких випадках він може бути помітним (видимі маркери), але здебільшого впроваджується непомітно — з метою автентифікації, фіксації часу створення або встановлення приналежності до певного джерела.

Отже, ключова відмінність між криптографією, стеганографією та цифровими водяними знаками полягає у їх функціональному призначенні: криптографія приховує зміст, стеганографія — сам факт передачі, а цифрове маркування — слугує для підтвердження власності або контролю цілісності. Вибір між ними залежить від конкретного сценарію: у деяких випадках вони можуть застосовуватися окремо, в інших — у комбінації для посилення загального рівня інформаційної безпеки [13, 14]. Більш детально про відмінності стеганографії від криптографії та цифрових водяних знаків можна ознайомитись на табл. 1.3.

Таблиця 1.3

Відмінності стеганографії від криптографії та цифрових водяних знаків

Критерії Порівняння	**Стеганографія**	**Криптографія**	**Цифрові водяні знаки**
1	**2**	**3**	**4**
Наявність зашифрованого контенту	Непомітна: документ виглядає як звичайний, без явних ознак прихованої інформації	Очевидна: шифротекст сигналізує про захищені дані	Може бути як помітним, так і непомітним, залежно від реалізації

Продовження табл. 1.3

1	2	3	4
Стійкість до обробки	Схильна до втрат при форматних змінах або перетвореннях	Залежить від алгоритму, при зміні структури втрачає придатність	Висока стійкість до стиснення, копіювання, перетворення
Помітність для сторонніх осіб	Повна непомітність при якісному впровадженні	Помітна наявність шифрування	Частково помітна або повністю прихована залежно від методу
Можливість поєднання з іншими	Може використовуватись разом із криптографією	Часто поєднується зі стеганографією для подвійного захисту	Може співіснувати з іншими методами захисту
Приклад використання	Вставлення прихованого повідомлення у Word-документ або зображення	Передача зашифрованого листа електронною поштою	Вбудований водяний знак у фото або відео для підтвердження авторства

Висновки за розділом 1

У розділі 1 було проаналізовано теоретичні основи стеганографії як засобу забезпечення інформаційної непомітності в цифрових середовищах. Її принципова особливість полягає у маскуванні самого факту наявності повідомлення, що кардинально відрізняє її від криптографії, де захист

забезпечується шляхом перетворення вмісту. У контексті інформаційної безпеки стеганографія виконує унікальну функцію прихованої комунікації, яка зберігає нейтральний вигляд цифрового носія та не викликає підозри навіть при зовнішньому аналізі.

Розглянута класифікація стеганографічних методів охоплює поділ за типами носіїв, формами взаємодії з даними та організацією схеми приховування. Залежно від обраного підходу, змінюються вимоги до структури документа, ступінь втручання в його вміст та потенційні вектори виявлення. У текстових форматах можливості впровадження інформації обмежені, однак саме їх активне використання в повсякденному обміні зумовлює стійкий інтерес до подальшого розвитку відповідних методик.

Визначальними критеріями якості стеганографічної системи залишаються непомітність, місткість і стійкість. Ефективність приховування базується на балансі між цими параметрами: чим більший обсяг інформації вбудовується, тим складніше зберегти її непоміченою; чим вищу стійкість необхідно забезпечити, тим складнішим стає механізм кодування. Усі ці фактори тісно пов’язані з особливостями носія та сценарієм його подальшого використання.

У контексті текстової стеганографії простежується тенденція до переходу від примітивних технік (наприклад, прихованих символів) до багаторівневих стратегій, які включають синтаксичні, лексичні та структурні трансформації. Використання форматів із гнучкою XML-структурою, зокрема DOCX, відкриває нові можливості для комбінованого приховування — не лише у видимому тексті, але й у службових компонентах документа. Актуальність набуває також застосування автоматизованих рішень, здатних адаптувати метод до зміни структури без втрати функціональності.

Встановлено принципові відмінності між стеганографією, криптографією та цифровим маркуванням. Якщо криптографія зосереджена на захисті змісту, а цифрові водяні знаки — на фіксації авторства та автентичності, то стеганографія орієнтована на збереження непомітності самої комунікації.

РОЗДІЛ 2
АНАЛІЗ МОЖЛИВОСТЕЙ СТЕГАНОГРАФІЇ У ДОКУМЕНТАХ MS WORD

2.1 Порівняння форматів DOC і DOCX як середовищ для приховування

Формати документів Microsoft Word, зокрема DOC і DOCX, становлять одне з найпоширеніших і водночас технологічно насичених текстових середовищ у структурі сучасного цифрового документообігу. Їх активне використання як у приватному листуванні, так і в офіційному адміністративному, навчальному чи корпоративному спілкуванні створює інформаційне середовище з високим рівнем довіри з боку кінцевого користувача. Саме ця обставина робить Word-документи надзвичайно зручним і природним контейнером для прихованого впровадження даних: приховані вставки не викликають підозри вже самим фактом свого носія. При цьому, на відміну від графічних або аудіооб’єктів, текстові документи рідше піддаються глибокому технічному аналізу, що додатково знижує ймовірність виявлення стеганографічного вмісту.

Формат DOC, що застосовувався до переходу Microsoft Office на відкриту архітектуру Open XML у 2007 році, є бінарною структурою закритого типу з низьким рівнем формалізації. У такому файлі інформація зберігається у вигляді послідовностей байтів без чітко визначених структурно-семантичних меж, що значно ускладнює як сторонній аналіз, так і розробку універсальних інструментів для її стеганографічного використання. Водночас саме ця непрозорість дозволяє використовувати такі файли для приховування даних у вигляді модифікацій службових ділянок, невикористаних байтів, спеціальних маркерів або навіть модифікацій форматуючих конструкцій, які не впливають на відображення змісту. Втім, через високий рівень внутрішньої складності та нестачу

документації будь-яке втручання в DOC-файл пов’язане з ризиком порушення його сумісності з офісними додатками, особливо у випадку повторного збереження, відкриття в інших версіях Word або конвертації.

На відміну від цього, формат DOCX, що реалізує принципи відкритої XML-структури, надає значно більше можливостей для цілеспрямованого й контрольованого приховування інформації. Документ у цьому форматі є ZIP-архівом, усередині якого знаходяться окремі XML-файли, відповідальні за різні функціональні компоненти документа — текстове наповнення, стилі, таблиці, об’єкти, метадані, службові параметри тощо. Така модульна організація дозволяє точно локалізувати місце приховування даних і при необхідності розмежовувати зони видимого й невидимого впливу. Крім того, структура DOCX підтримує програмний доступ через стандартні бібліотеки (наприклад, Open XML SDK або Python-бібліотеки для обробки DOCX), що відкриває можливості для створення автоматизованих систем приховування та витягування даних [15].

Одним із найбільш перспективних напрямів у цьому контексті є використання службових структурних елементів документа, які зберігаються незалежно від основного тексту, але входять до складу файлу під час збереження. До таких компонентів належать: властивості документа (автор, заголовок, ключові слова, час створення), користувацькі поля, приховані текстові блоки, коментарі, елементи колонтитулів, об’єкти гіперпосилань, а також елементи w:instrText чи w:fldSimple, що формують динамічні поля. Зміни в цих ділянках не порушують загальну структуру тексту, не впливають на його візуальне представлення і, що найважливіше, не фіксуються звичайними засобами перевірки формату або орфографії.

2.2 Методи приховування інформації в текстовому документі

У межах текстових документів стеганографічні методи реалізуються через вбудовування прихованих даних у ті структурні, форматні або службові елементи, які залишаються непомітними під час звичайного перегляду, проте

зберігаються у складі документа. Формати на зразок DOCX, що мають модульну XML-архітектуру, відкривають широкий спектр технічних можливостей для такого приховування — від роботи з символьним рівнем до маніпулювання структурними параметрами документа. Ключовим завданням при цьому є забезпечення непомітності змін та стійкості прихованої інформації до втрати у процесі редагування, збереження або конвертації.

Сутність методів полягає у мінімальному втручанні в основний вміст документа — приховані вставки повинні не порушувати смислову, граматичну чи візуальну цілісність. Усі зміни, як правило, відбуваються або на рівні технічної структури, яка недоступна пересічному користувачеві, або у форматних параметрах, що не мають вираженого графічного ефекту. Це дозволяє органічно інтегрувати приховані повідомлення навіть у документи, що виглядають цілком типовими за своїм оформленням та змістом.

Залежно від реалізації, приховані дані можуть бути закодовані у структурі символів, форматуванні, стилях, службових властивостях документа чи навіть у його метаданих. Такі підходи здатні забезпечити достатню місткість для передачі коротких повідомлень або службової інформації, не викликаючи підозри з боку користувача чи систем перевірки. Водночас ефективність методів суттєво залежить від того, наскільки вдало обраний канал приховування поєднує стійкість до автоматичних трансформацій із мінімальним ризиком виявлення прихованої активності [16, 17].

2.2.1 Недруковані символи Unicode

Одним із найбільш непомітних і технічно зручних засобів приховування інформації в текстових документах є використання недрукованих символів з репертуару Unicode, зокрема таких, що мають нульову ширину. Ці символи не створюють графічного відображення на екрані й не впливають на візуальне представлення тексту під час звичайного перегляду, проте фізично присутні у кодовій структурі документа. Саме ця властивість робить їх зручним носієм

прихованих бітових послідовностей, які можуть бути вбудовані у текст без порушення його семантики або граматики.

До найпоширеніших символів цієї категорії належать Zero Width Space (U+200B), Zero Width Non-Joiner (U+200C), Zero Width Joiner (U+200D), а також менш відомі Control Character Formatting Codes, що використовуються в окремих мовах або в типографічних стандартах. У межах стеганографічного алгоритму ці символи можуть бути інтерпретовані як логічні "0" або "1", залежно від їхнього порядку, комбінації або присутності у певних позиціях між словами, абзацами чи літерами. Такий спосіб кодування забезпечує високий рівень непомітності навіть у текстах, що проходять побіжний ручний або автоматизований перегляд.

Важливою перевагою цієї техніки є її незалежність від мовного контексту чи лінгвістичних структур: приховане повідомлення не залежить від змісту тексту, а вбудовується в його символічний каркас. Це дозволяє реалізовувати універсальні алгоритми, які не потребують адаптації до конкретної мови чи стилістики документа. Окрім цього, недруковані символи зберігаються у структурі документа після збереження, пересилання та повторного відкриття, за умови що програмне середовище підтримує повний набір символів Unicode.

Водночас застосування таких символів має і низку обмежень. Зокрема, автоматичне очищення форматування або конвертація документа у спрощений формат (наприклад, .txt або HTML без відповідної обробки Unicode) може призвести до часткової або повної втрати прихованої інформації. Окрім цього, деякі офісні системи безпеки або системи перевірки тексту можуть видаляти або замінювати нестандартні символи в межах процедур оптимізації. Таким чином, ефективне використання недрукованих символів потребує врахування специфіки середовища, у якому створюється, зберігається й передається документ [18].

2.2.2 Форматування, стилі, колонтитули як канали приховування

Форматування тексту, система стилів, службові поля та елементи структури документа, зокрема колонтитули, можуть ефективно

використовуватись як канали для прихованого впровадження даних у текстовий документ. На відміну від методів, що працюють на рівні символів, ці підходи орієнтовані на макрорівень оформлення документа, де зміни, хоча й технічно помітні, не сприймаються як аномальні в умовах звичайного візуального перегляду.

Зміни у форматуванні — такі як варіативність міжрядкового інтервалу, мікрозміни відступів, кеглю, вирівнювання, кольору або підкреслення — можуть бути використані для кодування інформації у вигляді бітових послідовностей або маркерів. Наприклад, наявність певного стилю форматування або його специфічна конфігурація може відповідати певному значенню, що зчитується при обробці документа спеціалізованим скриптом. Оскільки такі зміни часто є органічною частиною текстового дизайну (наприклад, у випадку інструкцій, технічної документації чи шаблонів), вони рідко викликають підозру навіть у разі наявності множинних відмінностей по всьому документу.

Стилі, особливо ті, що створені вручну або скопійовані з інших документів, можуть мати приховані параметри, які не використовуються для відображення тексту, але залишаються в структурі файлу. Назви стилів, їх пріоритет, наявність певних внутрішніх атрибутів можуть бути використані як контейнер для передавання службових повідомлень. У форматі DOCX ці стилі представлені окремим XML-файлом (styles.xml), що дає змогу точно контролювати зміст прихованого каналу та розмежовувати його від основного текстового шару.

Додатковим рівнем стеганографічного використання є службові поля документа (Field codes), які можуть містити динамічні значення, посилання, формули або умовні вирази. Вставлені вручну або автоматично, такі поля можуть кодувати інформацію у вигляді змінних, логічних операторів або комбінацій команд, які не обов’язково відображаються у підсумковому документі. Наприклад, поле типу INCLUDETEXT може посилатися на зовнішній файл, який містить закодоване повідомлення, але це не буде очевидно для користувача без доступу до вихідного коду документа.

Колонтитули (верхні та нижні), які зазвичай містять лише допоміжну інформацію (номери сторінок, дати, заголовки), також є потенційним середовищем для стеганографічного впливу. Оскільки вони повторюються на кожній сторінці та мають власну XML-структуру (header.xml, footer.xml), у них можна інтегрувати фрагменти тексту, коментарі або стилі, що будуть зберігатися в документі, але можуть бути майже непомітними на фоні загального оформлення [19].

2.2.3 Метадані документа: властивості, поля, службові ознаки

Метадані текстового документа є важливим, хоч і часто ігнорованим, шаром цифрової структури, придатним для стеганографічного використання. Вони не відображаються безпосередньо в основному вмісті, але зберігаються у складі документа, супроводжуючи його при збереженні, пересиланні або обробці. Саме їхня службова природа та вторинна видимість для користувача робить їх зручним середовищем для прихованого впровадження інформації.

У форматі DOCX метадані організовані у вигляді окремих XML-файлів, які містять інформацію про властивості документа (docProps/core.xml, docProps/app.xml) та користувацькі поля (custom.xml). Серед стандартних метаданих можна виокремити такі поля, як заголовок, тема, ключові слова, автор, категорія, дата створення, останній редактор тощо. Усі ці поля можуть зберігати довільні текстові значення, які не впливають на відображення документа в інтерфейсі користувача. Відповідно, ці атрибути можуть містити приховане повідомлення без викликання підозри, особливо якщо їх значення стилістично схожі на типові службові описи.

Крім стандартних, DOCX підтримує створення користувацьких властивостей — параметрів, які задаються вручну або автоматизовано і можуть мати будь-яке ім’я та значення. Вони зберігаються в окремому блоці й не мають прямого відображення у вмісті документа, однак залишаються доступними через інтерфейс властивостей або за допомогою відповідних API-запитів. Це дозволяє

приховувати дані в умовно «легітимному» середовищі, яке не зазнає змін при редагуванні тексту або форматування [3].

Окремий інтерес становлять службові ознаки документа, які генеруються текстовим процесором автоматично — такі як унікальні ідентифікатори, поля перегляду, мітки змін, залишки попередніх редагувань, прив’язки до шаблонів тощо. Частина з них зберігається у службових файлах структури (settings.xml, webSettings.xml, word/_rels), і може бути використана для вбудовування маркерів, ідентифікаційних кодів або навіть зашифрованих повідомлень. У більшості випадків ці дані не переглядаються користувачем без спеціальних засобів, але зберігаються протягом усього життєвого циклу документа [20].

2.2.4 Білі пробіли як носії прихованої інформації: типи, коди, особливості застосування

У контексті текстової стеганографії білі пробіли відіграють роль простого, проте ефективного засобу прихованого впровадження інформації. До цієї категорії належать символи, які не мають власного графічного представлення або не створюють помітного візуального ефекту у тексті, проте зберігаються у його символьній структурі. Завдяки цій особливості, білі пробіли активно використовуються як маркери, носії бітової інформації або умовні розділювачі, що не порушують логіки сприйняття документа користувачем.

Найпоширенішим є звичайний пробіл (ASCII-код 32, Unicode U+0020), який використовується для відділення слів у реченні. Поряд із ним, у документах застосовуються нерозривні пробіли, табуляції, символи переведення рядка. Ці символи можуть вставлятися між літерами, словами або абзацами, не змінюючи при цьому ані розміру, ані структури відображення тексту. Зокрема, Zero Width Space є майже ідеальним носієм прихованої інформації — він повністю невидимий у тексті, не викликає перенесення рядка й не порушує розмітку документа.

Стеганографічні методи, що базуються на використанні білих пробілів, передбачають, як правило, створення бінарного коду за допомогою чергування або наявності певних типів символів у визначених позиціях тексту. Наприклад, вставлення Zero Width Space може інтерпретуватися як логічна одиниця, а його відсутність — як нуль; або ж комбінація символів з різними кодами (наприклад, пробіл та нерозривний пробіл) використовується як двійкова система. Такий підхід є надзвичайно зручним у документах, що мають значний обсяг і форматовану структуру — приховані символи важко виявити без спеціального аналізу або перегляду у шістнадцятковому чи XML-представленні. Більш детально з інформацією про недруковані символи можна ознайомитись у табл. 2.1.

Незважаючи на очевидну привабливість такого способу, його ефективність залежить від здатності символів зберігатися при редагуванні, конвертації або пересиланні файлу. Автоматичні процеси очищення форматування, оновлення стилів або збереження документа в іншому форматі можуть призвести до часткової або повної втрати прихованої інформації. Особливо вразливими є символи з нульовою шириною, оскільки деякі текстові процесори не зберігають їх під час автоматичної оптимізації вмісту [21].

Таблиця 2.1

Недруковані символи

Назва символу	**Unicode**	**ASCII**	**Опис/призначення**
1	**2**	**3**	**4**
Пробіл (Space)	U+0020	32	Стандартний пробіл між словами.
Нерозривний пробіл (Non-breaking space)	U+00A0	160	Символ пробілу, який не допускає перенесення рядка.

Продовження табл. 2.1

1	2	3	4
Табуляція (Tab)	U+0009	9	Горизонтальна табуляція для вирівнювання тексту.
Зворотна табуляція (Backspace)	U+0008	8	Видалення попереднього символу (не використовується в тексті як пробіл).
Повернення каретки (Carriage return)	U+000D	13	Повертає курсор на початок рядка.
Переведення рядка (Line feed)	U+000A	10	Перехід на новий рядок без повернення каретки.
Zero Width Space	U+200B	-	Символ без ширини, не має візуального представлення.
Zero Width Non-Joiner	U+200C	-	Розділяє символи, запобігаючи їх злиттю (особливо в мовах із лігатурами).
Zero Width Joiner	U+200D	-	Поєднує символи, утворюючи лігатуру або комбіноване графічне представлення.

2.3 Проблематика виявлення та оцінки ефективності

Оцінка ефективності стеганографічних методів у текстовому середовищі нерозривно пов’язана з аналізом їхньої вразливості до виявлення — як вручну, так і за допомогою автоматизованих засобів. На відміну від мультимедійних форматів, де для детекції можуть застосовуватись спектральні або статистичні

методи, у тексті такі підходи потребують адаптації до структурно-семантичних особливостей документа. Це зумовлює специфічну складність розробки засобів стеганалізу, що мають виявляти приховану інформацію без доступу до еталонного (оригінального) документа.

Основною проблемою є те, що більшість стеганографічних вставок у тексті використовують легітимні ресурси структури документа — форматування, стилі, службові поля або недруковані символи, — які самі по собі не є ознаками порушення. Відтак, задача виявлення прихованої інформації перетворюється на аналіз імовірних аномалій у межах допустимої норми — наприклад, надмірне повторення певних форматних конструкцій, нетипова структура стилів, нестандартна щільність розміщення символів або наявність неочікуваних службових атрибутів. При цьому результати такого аналізу носять імовірнісний характер і потребують додаткової інтерпретації [22].

Іншим викликом є обмеженість існуючих інструментів для стеганалізу саме текстових документів. Більшість доступних засобів розроблені для роботи з мультимедійним контентом (зображеннями, аудіо, відео) і не підтримують специфіку текстових форматів, зокрема багатошарової структури DOCX або обробки нестандартних символів Unicode. Розробка повноцінних засобів аналізу текстової стеганографії вимагає врахування як форматних, так і лінгвістичних аспектів, що потребує міждисциплінарного підходу та залучення методів обробки природної мови, структурного аналізу XML-файлів і профілювання типових моделей поведінки документів.

2.3.1 Типові ознаки прихованих вставок і способи їх детекції

Ознаки наявності прихованих вставок у текстовому документі зазвичай не є однозначно визначеними, оскільки більшість стеганографічних методів навмисно працює в межах допустимого або навіть очікуваного форматного чи структурного шаблону. Проте при детальному аналізі документа, особливо в структурованому форматі DOCX, можливо виявити певні непрямі

характеристики, які можуть свідчити про потенційну присутність прихованої інформації.

Одним із ключових індикаторів є аномальна щільність або послідовність символів, зокрема використання недрукованих елементів Unicode. У випадках, коли такі символи вставлені у великій кількості або з відчутною регулярністю, вони можуть порушувати статистичну рівномірність тексту, що виявляється за допомогою символьного аналізу. Подібно до цього, надмірна кількість стилів або їх неочікуване чергування може свідчити про приховане кодування в параметрах форматування. Зокрема, підозру можуть викликати стилі, які не використовуються у вмісті, але містяться у структурі документа з унікальними назвами або зміненою конфігурацією [23].

Ще однією типовою ознакою є наявність великої кількості службових полів або нестандартних комбінацій XML-елементів у внутрішній структурі документа. Виявлення таких компонентів можливе при аналізі вмісту архіву DOCX або за допомогою спеціалізованих інструментів, які розпізнають та інтерпретують XML-теги, пов’язані з метаданими, гіперпосиланнями, макросами або полями динамічного тексту. Наявність незвичних або повторюваних тегів, а також параметрів, що не використовуються стандартними функціями документа, може вказувати на приховану активність.

Для виявлення таких ознак застосовуються кілька базових методів детекції. Серед них — структурний аналіз документа, що полягає у вивченні складових XML-файлів та пошуку нетипових елементів або неочікуваних змін у структурі стилів, форматів або службових полів. Крім того, використовуються методи символьного та частотного аналізу, які дозволяють виявляти нехарактерне розташування символів, порушення типових патернів або незвичну щільність міжсимвольного кодування.

У випадках, коли відомий оригінальний документ, застосовується диференційний аналіз, що полягає в побітовому або структурному порівнянні двох версій для виявлення непомітних змін. Проте в більшості реальних ситуацій такий підхід недоступний, що зумовлює актуальність досліджень у детекції,

орієнтованої на виявлення прихованих вставок без знання первинного стану документа [24].

2.4 Ризики та потенційні загрози використання стеганографії в текстових файлах

Хоча стеганографія вважається ефективним інструментом забезпечення конфіденційності комунікацій, її використання, особливо в текстових документах, супроводжується низкою суттєвих ризиків і викликів для інформаційної безпеки. Основна загроза полягає в можливості зловмисного застосування таких методів з метою передачі шкідливого вмісту, прихованих команд або витоку службової інформації у форматах, що здаються звичайними й легітимними.

У середовищах із високими вимогами до безпеки, таких як корпоративні або державні IT-інфраструктури, наявність прихованих даних у Word-документах може бути індикатором внутрішньої загрози або цілеспрямованої атаки. Особливу складність становить те, що вбудовані повідомлення можуть бути невидимими навіть для фахівців, якщо не застосовувати спеціалізованих інструментів стеганалізу. Це створює сприятливе підґрунтя для прихованого обміну інформацією без відома власника системи [25].

2.4.1 Використання стеганографії для маскування шкідливого коду

Одним із найнебезпечніших векторів зловмисного застосування текстової стеганографії є приховане впровадження шкідливого коду у внутрішню структуру документа. У контексті форматів DOC та DOCX, які підтримують елементи автоматизації (зокрема макрокоманди, поля з динамічними значеннями, зв’язки з зовнішніми джерелами), зловмисники отримують потужний інструмент для створення складних, стійких до виявлення загроз.

Найпоширенішим прикладом є використання макросів у поєднанні зі стеганографічним маскуванням: код може бути закладений у спеціальних полях (наприклад, IF, INCLUDETEXT, FORMTEXT), стилях або метаданих, а його запуск ініціюється при певних умовах — відкритті файлу, перемиканні перегляду або взаємодії з елементами інтерфейсу. Такі конструкції часто залишаються поза межами візуального контролю користувача, оскільки не відображаються безпосередньо в тексті або приховані в службових областях документа [26].

У структурі DOCX також існує можливість інтеграції посилань на зовнішні файли або сценарії (наприклад, через XML-атрибути в rels-файлах), які можуть виконуватись або підвантажуватись автоматично за певних умов. Подібні механізми часто застосовуються у фішингових атаках або при поширенні програм-шпигунів.

Окрему загрозу становлять так звані “steganomacro”-підходи, коли код розбивається на фрагменти, що вставляються у вигляді недрукованих символів, структурних змін або прихованих елементів, і лише під час обробки документа з використанням конкретного середовища (наприклад, VBA-парсера) об’єднується в робочий скрипт. Це робить виявлення такого вмісту особливо складним без глибокого технічного аналізу документа на рівні XML-структури або бінарних елементів [27,28].

2.5 Приклади використання текстової стеганографії в реальних сценаріях

Текстова стеганографія, попри свою технічну складність, активно застосовується у практичних випадках, де потрібне непомітне впровадження або передавання інформації. На відміну від мультимедійної стеганографії, текстові методи не вимагають великих обсягів даних чи спеціалізованих носіїв, що робить їх придатними для використання в умовах підвищеного контролю — наприклад,

у внутрішній корпоративній комунікації, державному документообігу або при обміні службовою інформацією через публічні канали [29].

Одним із найпоширеніших сценаріїв є маскування службових повідомлень у текстових документах, що передаються електронною поштою. У таких випадках стеганографічне кодування виконується за допомогою недрукованих символів або комбінацій стилів, які не змінюють вигляду документа, але можуть бути прочитані спеціалізованими засобами.

Іншим важливим вектором застосування є організація прихованих каналів обміну у середовищах з обмеженим використанням шифрування, де застосування криптографії є недопустимим або може бути розцінене як підозріле. У таких випадках текстова стеганографія слугує альтернативним способом передачі конфіденційної інформації.

Не менш актуальними є й зловмисні сценарії, зокрема використання текстових документів для прихованого впровадження шкідливого коду, команд або маркерів для наступного завантаження вірусних компонентів. Завдяки складній структурі DOCX-файлів, код може бути розміщено у таких ділянках, як метадані, колонтитули, поля, або навіть у макрокомандах, які активуються при відкритті [30].

Висновки за розділом 2

У другому розділі було проаналізовано можливості використання текстових документів Microsoft Word як середовища для прихованого впровадження інформації, а також пов’язані з цим ризики, обмеження та способи виявлення таких вставок. Особливості форматів DOC і DOCX визначають різний рівень придатності до стеганографії: DOCX, завдяки відкритій XML-архітектурі, надає значно ширші технічні можливості, тоді як бінарний DOC характеризується меншою гнучкістю, але вищою непрозорістю структури.

У першому підрозділі розкрито специфіку форматів DOC і DOCX як середовищ для реалізації прихованих вставок. Порівняно з застарілим DOC,

формат DOCX забезпечує розподілену модульну структуру, де дані зберігаються в окремих XML-компонентах. Це дозволяє впроваджувати стеганографічні дані в різні шари документа — від основного тексту до стилів, полів, метаданих і службових тегів. Проаналізовано переваги роботи з DOCX у контексті програмного доступу, автоматизації приховування та контролю структури.

Другий підрозділ був присвячений огляду методів приховування інформації в текстовому документі. Розглянуто принципи роботи з недрукованими символами, варіаціями форматування, стилями та службовими елементами Word-документів. Основна увага приділена тому, як ці елементи можуть бути модифіковані без помітного впливу на вміст або вигляд документа, а також наскільки вони стійкі до редагування, конвертації й автоматичної оптимізації, що часто виконується офісними платформами.

Третій підрозділ був присвячений виявленню типових ознак прихованих вставок у текстових документах. Проаналізовано, які структурні або символьні аномалії можуть свідчити про наявність стеганографічної інформації, зокрема використання недрукованих символів Unicode, надлишкових стилів або службових XML-елементів із нетиповими параметрами. Основна увага зосереджена на підходах до детекції — від структурного аналізу DOCX-файлів до частотного аналізу й порівняння версій документів. Наголошено на складності виявлення прихованих вставок без еталонного документа та потребі в розробці безеталонних методів виявлення стеганографічної активності.

РОЗДІЛ 3
РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ДЛЯ ПРИХОВАННЯ ІНФОРМАЦІЇ У НЕДРУКОВАНИХ СИМВОЛАХ WORD – ДОКУМЕНТІВ

3.1 Формулювання задачі та технічні вимоги до рішення

У межах розробки було поставлено завдання створити спеціалізований програмний засіб, здатний виконувати приховане впровадження довільного текстового повідомлення у структуру DOCX-документів з використанням недрукованих символів Unicode. Основна мета полягала у створенні такого механізму приховування, який забезпечував би збереження повної невидимості впроваджених даних для користувача під час стандартного перегляду або редагування файлу в офісних текстових процесорах. Водночас інструмент мав демонструвати достатню стійкість до типових змін структури документа — зокрема, тих, що відбуваються внаслідок повторного збереження, конвертації, фрагментарного редагування або копіювання вмісту в інші файли.

Функціональне ядро програмного засобу базується на двох взаємодоповнюючих модулях. Перший — модуль кодування — реалізує алгоритми перетворення вхідного тексту у послідовність невидимих символів, що відповідають стандарту Unicode (наприклад, Zero Width Space, Zero Width Joiner тощо), та здійснює їх вбудовування у вміст документа відповідно до заданої логіки впровадження. Така логіка може бути лінійною (послідовне вбудовування після певних текстових маркерів), псевдовипадковою (залежною від генератора псевдовипадкових чисел із фіксованим ключем), або структурною (з орієнтацією на логічні розділи документа — наприклад, таблиці, абзаци, службові блоки).

Другий компонент — модуль декодування — виконує зворотне завдання: аналізує структуру документа, ідентифікує вставлені недруковані символи, вилучає приховану послідовність та відновлює первинне повідомлення. У цьому

процесі не вимагається попереднє редагування файлу, вручну задані правила або додаткові зовнішні мітки. Завдяки цьому декодування може бути реалізовано автоматизовано — з мінімальною взаємодією з боку користувача.

Окремим напрямом розробки стало забезпечення широкої сумісності програмного рішення з різними реалізаціями формату DOCX, які генеруються популярними текстовими редакторами — Microsoft Word, LibreOffice Writer, Google Docs тощо. Це вимагало дотримання специфікацій Open XML та обережної роботи з XML-компонентами документа, які можуть мати відмінності залежно від редактора. У результаті програмний засіб було спроєктовано як незалежний від конкретного середовища генератор/декодер прихованої інформації.

Для зручності користувача реалізовано інтуїтивно зрозумілий інтерфейс, який дозволяє завантажувати файли, вводити або зчитувати повідомлення, а також перевіряти коректність виконаних операцій. Особливу увагу під час проєктування було приділено механізмам протидії виявленню вставок — зокрема, розосередженню символів, варіативності їх розміщення, уникненню повторюваних структур або статистичних аномалій, що могли б бути виявлені інструментами стеганалізу.

3.2 Програмна реалізація засобу стеганографічного приховування інформації у недрукованих символах «PanchenkoSteganography»

Програмний засіб *“PanchenkoSteganography”* було розроблено для реалізації прихованого впровадження текстової інформації у документи формату DOCX шляхом маніпулювання недрукованими символами, зокрема пробілами, яким призначаються спеціальні кольорові атрибути. Основною метою програми є забезпечення такої інтеграції даних, яка не змінює візуального вигляду документа під час його відкриття в стандартному текстовому редакторі, але дозволяє надійно відновити вміщене повідомлення шляхом технічного зчитування коду кольору пробілів.

Архітектура програмного засобу побудована на основі середовища .NET із застосуванням мови програмування C# та бібліотеки Open XML SDK, що надає доступ до структури DOCX-файлів на рівні XML-елементів. Програма має графічний інтерфейс, який дозволяє користувачеві завантажувати вихідний документ (що виступає контейнером), вводити повідомлення для приховування, запускати процес кодування, а також зчитувати приховані дані з іншого документа. Усі дії реалізовано через окремі функції, які забезпечують інкапсуляцію логіки взаємодії з файлом.

Процес кодування виконується функцією Embedd(), яка спочатку створює копію обраного документа, а далі аналізує його вміст на рівні абзаців і текстових елементів (Paragraph, Run). Під час проходження по кожному Run виконується пошук пробільних символів. Перед кожним пробілом фрагмент вихідного тексту розділяється на окремий текстовий елемент з відповідним збереженням стилістичних властивостей. Далі сам пробіл модифікується так, щоб його атрибут кольору (RunProperties.Color) кодував три байти з текстового повідомлення — через значення компонентів RGB. Ці значення можуть бути частково заповненими, якщо довжина повідомлення не кратна трьом, при цьому залишкові байти доповнюються нулями. Завдяки використанню властивості SpaceProcessingModeValues.Preserve текстові пробіли не зникають при збереженні документа. Усі модифіковані елементи зберігаються в новому документі (result.docx), що містить приховане повідомлення у вигляді колірного кодування пробілів.

Функція Extract() виконує обернену операцію: вона відкриває обраний файл, сканує всі елементи типу Run, що мають властивість кольору, та витягує з неї послідовність RGB-значень, які інтерпретує як байти. Отримані байти з'єднуються у масив і декодуються у текстове повідомлення за допомогою UTF-8. Це дає змогу повністю відновити приховану інформацію без необхідності редагування або ручного аналізу документа.

Алгоритм роботи можна описати наступним чином (Додаток А):

- Користувач натискає кнопку “Choose” у блоці Empty Container File, після чого відкривається діалог вибору файлу. Обирається DOCX-документ, що містить початковий текст і слугуватиме контейнером. У полі “Secret message to hide” користувач вводить повідомлення, яке потрібно приховати. Приклад роботи зображено на рис. 3.1;

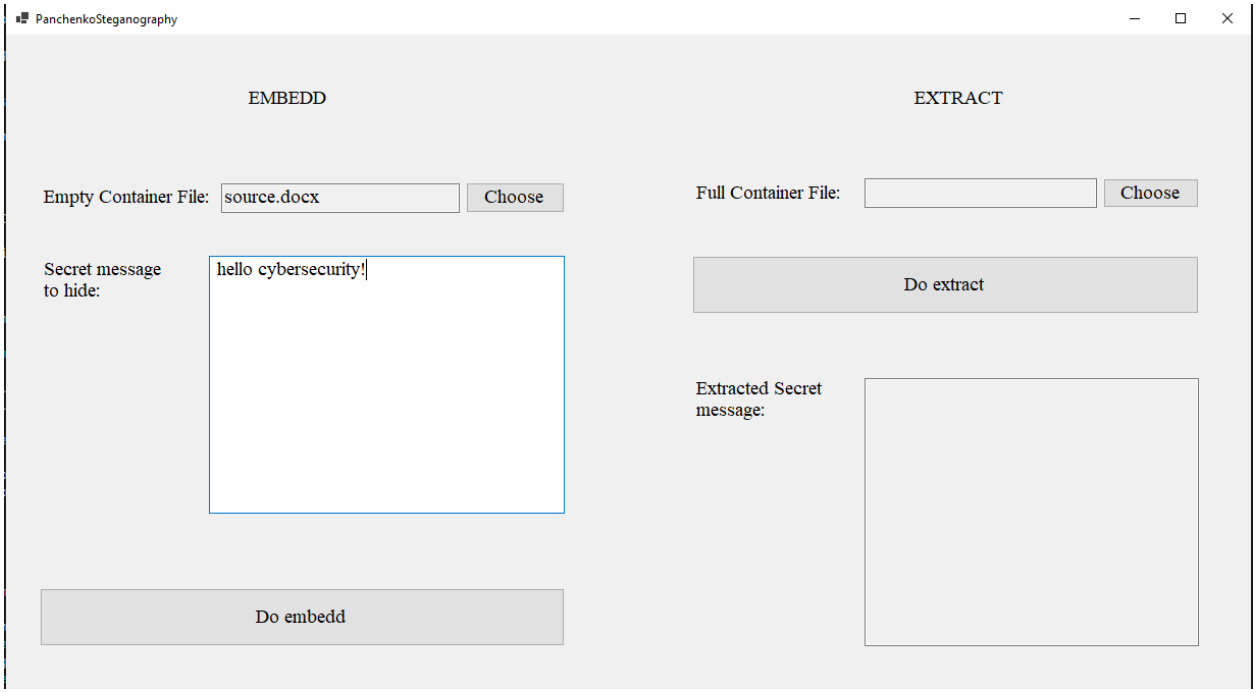


Рисунок 3.1 – Процесс приховування тексту у файлі.

Натискається кнопка “Do embedd”. У цей момент запускається метод Embedd, який:

- Копіює обраний файл у новий файл result.docx;
- Відкриває документ і перебирає всі абзаци та текстові блоки (Run) у тілі документа;
- Шукає пробільні символи (space, tab, \n, тощо) в тексті;
- Перед кожним знайденим пробілом вставляє частину тексту без змін, додає пробіл з кольором, що кодує 3 символи повідомлення у форматі RGB;

Таким чином, повідомлення кодується у вигляді невидимих кольорових пробілів. Після завершення кодування з’являється повідомлення “Secret message

was successfully embedded!”, інтерфейс очищається. Можна ознайомитись з результатами на рис. 3.2.

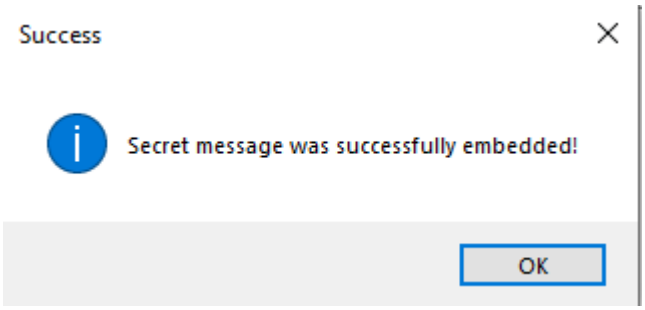


Рисунок 3.2 – Повідомлення успішно приховано.

Користувач натискає кнопку “Choose” у блоці Full Container File та обирає документ, у якому було приховано повідомлення. Шлях до файлу зберігається у змінній. Натискається кнопка “Do extract”. З приницпом роботи можна ознайомитись на рис. 3.3. Запускається метод Extract, який:

- Відкриває DOCX-файл лише для читання;
- Проходить усі Run-елементи в тілі документа;
- Перевіряє кожен елемент на наявність кольору;
- Атрибут Color має довжину 6 символів (RGB):
    - розбиває значення на три компоненти (червоний, зелений, синій);
    - перетворює кожен компонент у байт і додає до списку

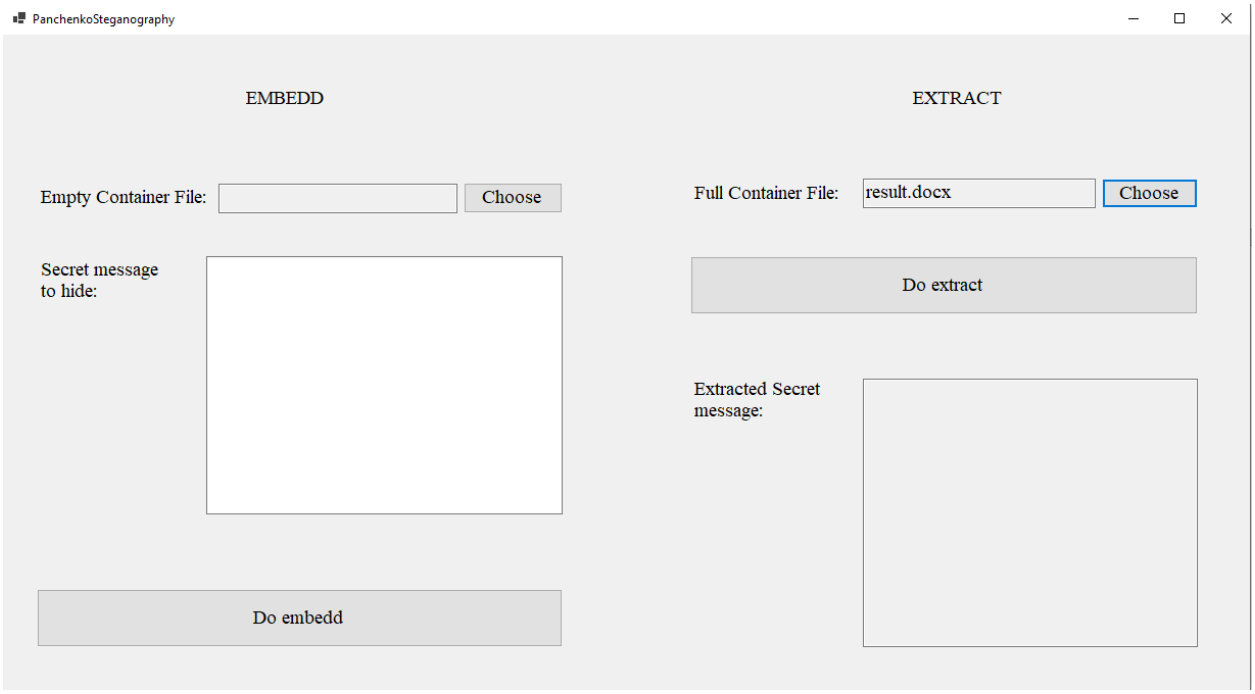


Рисунок 3.3 – Процесс витягання зашифрованого повідомлення з тексту.

Зібрані байти декодуються у текст (UTF-8), результат виводиться у поле “Extracted Secret message”. З’являється повідомлення “Secret message was successfully extracted!”. Ознайомитись з результатом можна на рис. 3.4.

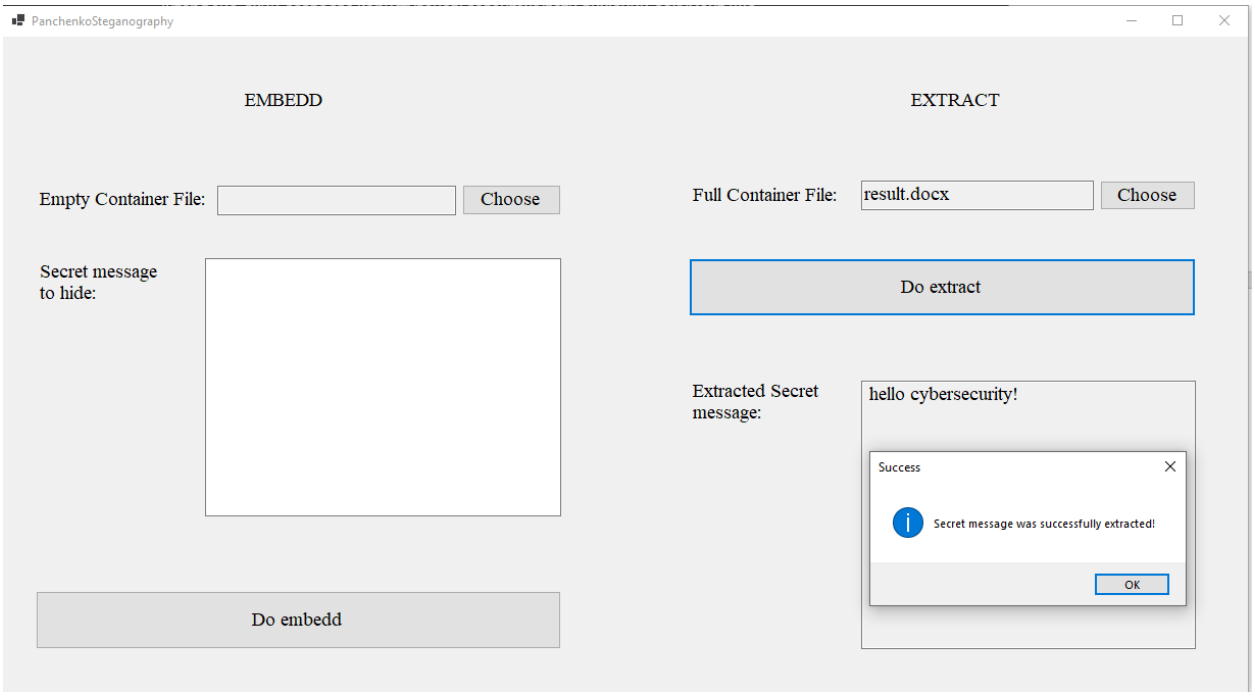


Рисунок 3.4 – Повідомлення успішно витягнуто.

3.3 Оцінка ефективності та обґрунтування подальших напрямів удосконалення

Оцінювання ефективності реалізованого програмного засобу здійснювалося за багатофакторним підходом, який включав як якісні, так і кількісні характеристики прихованого впровадження інформації в структуру DOCX-документів. Основну увагу було приділено наступним аспектам: ступеню візуальної непомітності зашифрованого повідомлення, стійкості до змін документа в процесі типового користувацького редагування, відповідності структури файлу специфікаціям OpenXML після модифікацій, а також коректності декодування вбудованих даних на різних етапах експлуатації. Додатково враховувалися параметри швидкодії, сумісності з офісними редакторами та захищеності прихованого каналу від випадкового виявлення.

Результати функціонального тестування підтвердили відповідність засобу ключовим вимогам: застосування недрукованих символів (зокрема, Zero Width Space, U+200B) у поєднанні з маніпулюванням колірними параметрами символів не викликало жодних відображуваних змін у текстовому представленні документа в середовищі Microsoft Word. Це свідчить про ефективну реалізацію базового принципу стеганографії — приховування самого факту наявності додаткової інформації. Повідомлення залишалося цілісним навіть після збереження документа в іншому офісному середовищі, часткового редагування вмісту або копіювання окремих фрагментів. Відповідно, стійкість до базових модифікацій була досягнута без залучення надлишкових засобів обробки.

Швидкодія алгоритмів як на етапі впровадження, так і при зчитуванні прихованого тексту виявилася задовільною. Навіть при обробці об’ємних документів (розміром понад 500 сторінок) затримки залишалися в межах кількох секунд, що є прийнятним результатом для локального десктопного рішення. Алгоритм декодування продемонстрував високу надійність — навіть за умов часткового пошкодження оформлення або втрати окремих стилістичних

параметрів, декодований текст відновлювався повністю, якщо збережено принаймні основну частину контейнера.

З метою подальшого вдосконалення програмного засобу доцільно розглянути декілька перспективних напрямів розвитку. Зокрема:

- Розширення типів каналів прихованого впровадження, зокрема за рахунок використання службових полів Word (Field Codes), модифікації структур XML-файлів (наприклад, settings.xml, styles.xml) або метаданих користувача;
- Вбудована криптографічна обробка перед передачею повідомлення до каналу стеганографії, що дозволить підвищити захист даних у випадку виявлення прихованого вмісту;
- Створення стеганалізатора — модуля виявлення слідів приховування, який дозволить оцінити ризики розкриття або виявлення інформації за допомогою статистичного або структурного аналізу документа;
- Розширення підтримуваних форматів, зокрема ODT (OpenDocument), RTF, HTML та PDF, що дозволить адаптувати засіб до ширшого спектру документних систем;
- Візуалізація прихованих повідомлень, наприклад, через автоматичне генерування QR-кодів або інтеграцію з засобами звітності, що розширить можливості використання в захищеному документообігу.

Таким чином, створене рішення демонструє не лише практичну ефективність у рамках поставленої задачі, а й має високий потенціал для подальшої адаптації до вимог реальних інформаційно-телекомунікаційних середовищ, де безпека й непомітність обміну даними залишаються критичними параметрами.

Висновки за розділом 3

У третьому розділі було створено спеціалізований програмний засіб «PanchenkoSteganography», який реалізує стеганографічне приховування

інформації у недрукованих символах DOCX-документів. Основною особливістю програми є використання Unicode-символів, таких як Zero Width Space, а також кольорових атрибутів пробілів, що дозволяє вбудовувати приховані повідомлення у текст без зміни його зовнішнього вигляду при відкритті в офісних редакторах. Застосована технологія забезпечує повну невидимість прихованої інформації для пересічного користувача та високу стійкість до змін, які можуть виникати під час редагування, копіювання або повторного збереження документа.

Функціонально програма складається з двох основних модулів: кодування та декодування. Перший дозволяє користувачеві вбудувати текстове повідомлення у документ шляхом кодування байтів повідомлення у вигляді RGB-компонентів кольору пробілів, які не змінюють вигляд тексту. Другий модуль зчитує ці кольори з документа та відновлює початкове повідомлення без необхідності ручного втручання або попереднього форматування файлу. Програму реалізовано мовою C# з використанням бібліотеки Open XML SDK, що забезпечує доступ до внутрішньої XML-структури документів формату DOCX. Інтерфейс користувача є простим і зрозумілим, дозволяючи завантажити контейнер-файл, ввести або зчитати приховане повідомлення, та миттєво отримати зворотний результат.

Проведене тестування підтвердило, що застосування недрукованих символів і кольорового кодування не порушує структури документа та не викликає візуальних змін у його відображенні, зберігаючи повну сумісність з популярними офісними пакетами — Microsoft Word, LibreOffice та Google Docs. Повідомлення зберігається навіть після часткового редагування або конвертації документа. Програма демонструє високу швидкодію навіть на документах великого обсягу, а алгоритм декодування залишається надійним навіть у разі втрати стилістичних атрибутів тексту.

У підсумку, програмний засіб успішно виконує поставлені завдання щодо прихованого зберігання інформації у DOCX-документах та демонструє практичну ефективність, зберігаючи невидимість та стійкість прихованих даних.

Надалі доцільно розширити функціональність за рахунок підтримки нових форматів файлів (ODT, PDF, RTF), інтеграції криптографічного захисту повідомлень, використання додаткових каналів приховування (наприклад, метаданих або службових полів Word), а також впровадження модуля стеганалізу для оцінки ризиків виявлення. Таким чином, створений інструмент є не лише функціонально завершеним рішенням для стеганографії у текстових документах, а й має значний потенціал для розвитку в умовах сучасних інформаційно-телекомунікаційних систем, де конфіденційність і прихованість обміну даними мають критичне значення.

ВИСНОВКИ

У даній кваліфікаційній роботі було реалізовано програмний засіб, призначений для прихованого впровадження текстової інформації в структуру документів формату DOCX з використанням методів текстової стеганографії на основі недрукованих символів. Розроблене рішення дозволяє забезпечити збереження повідомлення в прихованому вигляді без порушення візуального оформлення документа, що підтверджує його прикладну цінність в умовах зростаючих вимог до конфіденційності в інформаційно-телекомунікаційних системах.

У першій частині роботи було систематизовано теоретичні основи стеганографії як методу прихованого впровадження даних. Розглянуто ключові принципи, класифікацію методів, технічні вимоги до систем приховування інформації, а також проведено порівняльний аналіз між стеганографією, криптографією та цифровими водяними знаками. Також окреслено сучасні напрями розвитку текстових методів та їхні потенційні ризики й загрози.

У другій частині увагу зосереджено на аналізі можливостей використання текстової стеганографії у документах Microsoft Word. Проведено порівняння форматів DOC і DOCX, описано конкретні механізми приховування на рівні символів Unicode, форматування, метаданих і службових структур документа. Наведено приклади реального застосування стеганографії в практиці цифрових атак і захисту даних, що дало змогу обґрунтувати актуальність програмної реалізації.

У третій частині безпосередньо реалізовано та протестовано програмний інструмент, що здійснює кодування й декодування повідомлень через приховану зміну властивостей символів у DOCX-документі. Було описано архітектуру системи, розроблені алгоритми, інтерфейс користувача, а також виконано оцінку ефективності впровадженого рішення за критеріями невидимості, стійкості до змін, швидкодії та універсальності. Запропоновано шляхи подальшого розвитку,

включаючи розширення функціональності та підвищення рівня інформаційної безпеки.

Загалом, дана кваліфікаційна робота демонструє практичну реалізацію стеганографічного підходу до прихованого збереження даних у текстових документах, що може бути корисним як у сфері інформаційного захисту, так і в умовах, де потрібна конфіденційна передача інформації без викриття самого факту її існування. Робота поєднує теоретичне підґрунтя, аналітичний розгляд ризиків і сучасну інженерну реалізацію, що свідчить про її актуальність та практичну значущість.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Актуальність використання прихованої передачі даних у відеофайлах / Є. Ю. Катаєва, А. Г. Ребріков // Управління розвитком складних систем. - 2021. - Вип. 46. - С. 48-54. [Електронний ресурс] – Режим доступу: http://nbuv.gov.ua/UJRN/Urss_2021_46_9

2. Методика оцінки стеганографічних методів приховування інформації в зображеннях / О. Туровський та ін. Інфокомунікаційні та комп’ютерні технології. 2022. Т. 2, № 02. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.36994/2788-5518-2021-02-02-23.

3. Сучасні стеганографічні методи захисту інформації / О. І. Стасюк та ін. 2011. Т. 13, № 1 (50). [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.18372/2410-7840.13.1994.

4. D. K. Steganographic algorithms and stegoanalysis based on classical methods and neural networks. 2023. Т. 2, № 37. С. 42–53. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.31474/1996-1588-2023-2-37-42-53.

5. Umanskiy O. G. Review of steganographic methods and their application in securing banking information systems. 2024. Т. 46, № 4. С. 87–111. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.15407/emodel.46.04.087.

6. Шляхетський А. А. Стеганографія із застосуванням графічних зображень [Електронний ресурс] – Режим доступу до ресурсу: http://essuir.sumdu.edu.ua/handle/123456789/65613.

7. RasoolMahmood N., Abdul Azeez Mohammad A., Nesir Rasool Z. Public Key Steganography. International Journal of Computer Applications. 2014. Vol. 100, no. 8. P. 29–32. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.5120/17547-8143.

8. Steganography Security: Principle and Practice / Y. Ke et al. IEEE Access. 2018. Vol. 6. P. 73009–73022. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.1109/access.2018.2881680.

9. RasoolMahmood N., Abdul Azeez Mohammad A., Nesir Rasool Z. Public Key Steganography. International Journal of Computer Applications. 2014. Vol. 100, no. 8. P. 29–32. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.5120/17547-8143.

10. Ms. Halima Abbas Ahmed. Comprehensive Review of Cryptography and Steganography Algorithms. Journal of Information Systems Engineering and Management. 2025. Vol. 10, no. 29s. P. 211–228. [Електронний ресурс] – Режим доступу до ресурсу:https://doi.org/10.52783/jisem.v10i29s.4471.

11. Khaldi A. Diffie-Hellman Key Exchange through Steganographied Images. Law, State and Telecommunications Review. 2018. Vol. 10, no. 1. P. 147–160. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.26512/lstr.v10i1.21504.

12. Ghoul S., Sulaiman R., Shukur Z. A Review on Security Techniques in Image Steganography. International Journal of Advanced Computer Science and Applications. 2023. Vol. 14, no. 6. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.14569/ijacsa.2023.0140640.

13. Чорна А. В., Смірнов О. А., Мелешко Є. В. Розробка програмного забезпечення методу приховання інформації у мережі на основі стеганографії : thesis. 2012. [Електронний ресурс] – Режим доступу до ресурсу: http://dspace.kntu.kr.ua/jspui/handle/123456789/4337.

14. Video Based Steganography (Motion Vector Steganography) / Della Patresya Sitohang et al. Jurnal Teknik Indonesia. 2023. Vol. 2, no. 01. P. 33–38. . [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.58471/ju-ti.v2i01.662.

15. Video steganography: recent advances and challenges / J. Kunhoth et al. Multimedia Tools and Applications. 2023. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.1007/s11042-023-14844-w.

16. M K. Visual Cryptography and Steganography Methods Review. International Journal on Recent and Innovation Trends in Computing and

Communication. 2015. Vol. 3, no. 4. P. 1927–1930. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.17762/ijritcc2321-8169.150437.

17. Wavelet Methods in Steganography / E. S. Yakovleva et al. KnE Engineering. 2018. Vol. 3, no. 4. P. 318. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.18502/keg.v3i4.2255.

20. Stoyanova V. T. Steganography System Using LSB Methods. SSRN Electronic Journal. 2018. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.2139/ssrn.3283729.

21. Al-Hamammi A., M. H. A.-H. Proving Poverty of Steganography System. Information Technology Journal. 2005. Vol. 4, no. 3. P. 284–288. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.3923/itj.2005.284.288.

22. Ahmed A. M., Rashid I. M. Review of Steganography Algorithms. International Journal of Scientific Research in Science, Engineering and Technology. 2020. P. 01–09. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.32628/ijsrset20726.

23. Piotrowski Z. Analysis of selected steganography algorithms. ELEKTRONIKA - KONSTRUKCJE, TECHNOLOGIE, ZASTOSOWANIA. 2015. Vol. 1, no. 4. P. 33–36. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.15199/13.2015.4.6.

24. Warkentin M., Schmidt M. B., Bekkering E. Steganography and Steganalysis. Enterprise Information Systems Assurance and System Security. P. 287–294. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.4018/978-1-59140-911-3.ch018.

25. El Abbadi N. New Algorithm for Text in Text Steganography. Journal of Al-Rafidain University College For Sciences. 2021. No. 2. P. 99–112. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.55562/jrucs.v23i2.483.

26. Text, Image and Audio Steganography / M. T. N. Aruna et al. International Journal for Research in Applied Science and Engineering Technology. 2023. Vol. 11, no. 4. P. 4435–4439. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.22214/ijraset.2023.51091.

27. Cvejic N. (. Algorithms for audio watermarking and steganography : doctoral thesis. 2004. [Електронний ресурс] – Режим доступу до ресурсу: http://urn.fi/urn:isbn:9514273842.

28. D. K. STEGANOGRAPHIC ALGORITHMS AND STEGOANALYSIS BASED ON CLASSICAL METHODS AND NEURAL NETWORKS. Scientific papers of Donetsk National Technical University. Series: Informatics, Cybernetics and Computer Science. 2023. Vol. 2, no. 37. P. 42–53. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.31474/1996-1588-2023-2-37-42-53.

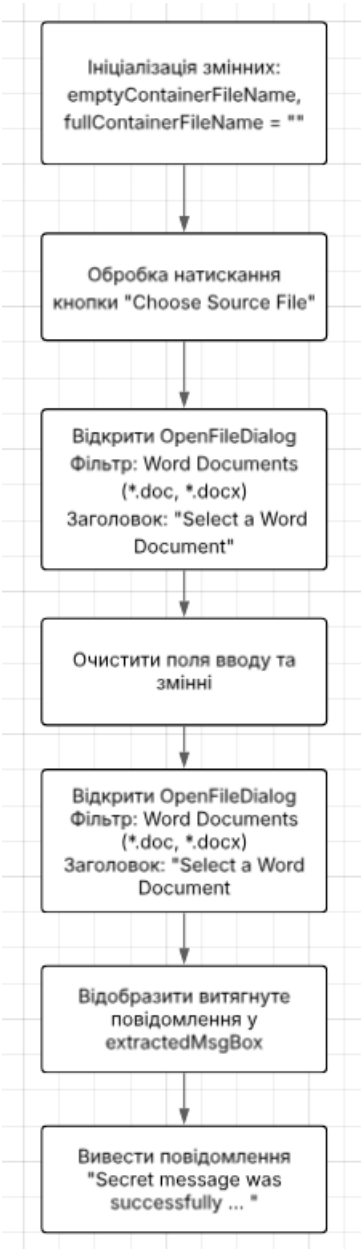
29. Шляхетський А. А. Стеганографія із застосуванням графічних зображень: thesis. 2017. [Електронний ресурс] – Режим доступу до ресурсу: http://essuir.sumdu.edu.ua/handle/123456789/65613.

30. AL-Durrah Q., AL-Assadi T. A. A Survey on Video Steganography Based on the Techniques and Evaluation Metrics. 2023 Second International Conference on Advanced Computer Applications (ACA), Misan, Iraq, 27–28 February 2023. 2023. [Електронний ресурс] – Режим доступу до ресурсу: https://doi.org/10.1109/aca57612.2023.10346630.

ДОДАТКИ

Додаток А

Графічне відображення структури програмного рішення



Додаток Б

Лістинг програмного рішення «PanchenkoSteganography»

```
using static System.Net.Mime.MediaTypeNames;
using System.Text;
using System.Drawing;
using System.Text;
using DocumentFormat.OpenXml;
using DocumentFormat.OpenXml.Packaging;
using DocumentFormat.OpenXml.Presentation;
using DocumentFormat.OpenXml.Wordprocessing;
using Color = DocumentFormat.OpenXml.Wordprocessing.Color;
using Text = DocumentFormat.OpenXml.Wordprocessing.Text;

namespace PanchenkoSteganography
{
   public partial class Form1 : Form
   {
      string emptyContainerFileName, fullContainerFileName;

      public Form1()
      {
         InitializeComponent();

         emptyContainerFileName = fullContainerFileName = string.Empty;
      }

      private void chooceSoursefileBtn_Click(object sender, EventArgs e)
      {
```

```
        using (OpenFileDialog openFileDialog = new OpenFileDialog())
        {
            openFileDialog.Filter = "Word Documents (*.doc;*.docx)|*.doc;*.docx";
            openFileDialog.Title = "Select a Word Document";

            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                emptyContainerFileName = openFileDialog.FileName;
                emptyContainerNameBox.Text =
Path.GetFileName(emptyContainerFileName);
            }
        }
    }

    private void embeedBtn_Click(object sender, EventArgs e)
    {
        if(emptyContainerFileName == string.Empty)
            MessageBox.Show("Choose empty container file", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        else if(msgToEmbeedBox.Text.Length == 0)
            MessageBox.Show("Enter secret text", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        else
        {
            Embedd(msgToEmbeedBox.Text, emptyContainerFileName);

            emptyContainerNameBox.Text = msgToEmbeedBox.Text =
emptyContainerFileName = string.Empty;
            MessageBox.Show("Secret message was successfully embedded!",
"Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
```

```
        }
    }

    private void chooseFullContainerBtn_Click(object sender, EventArgs e)
    {
        using (OpenFileDialog openFileDialog = new OpenFileDialog())
        {
            openFileDialog.Filter = "Word Documents (*.doc;*.docx)|*.doc;*.docx";
            openFileDialog.Title = "Select a Word Document";

            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                fullContainerFileName = openFileDialog.FileName;
                fullContainerNameBox.Text =
Path.GetFileName(fullContainerFileName);

                extractedMsgBox.Text = string.Empty;
            }
        }
    }

    private void extractBtn_Click(object sender, EventArgs e)
    {
        if (fullContainerFileName == string.Empty)
            MessageBox.Show("Choose empty container file", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        else
        {
            extractedMsgBox.Text = Extract(fullContainerFileName);
```

```
            MessageBox.Show("Secret message was successfully extracted!",
"Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }

    void Embedd(string secret, string emptyContainer, string fullContainer =
"./result.docx")
    {
        if (File.Exists(fullContainer))
            File.Delete(fullContainer);

        File.Copy(emptyContainer, fullContainer);
        using var wordDoc = WordprocessingDocument.Open(fullContainer, true);

        var currentByte = 0;
        var body = wordDoc.MainDocumentPart!.Document.Body!;
        var newElements = new List<OpenXmlElement>();

        foreach (var element in body.Elements())
        {
            if (currentByte >= secret.Length)
            {
                newElements.Add(element.CloneNode(true));
                continue;
            }

            if (element is not Paragraph paragraph)
            {
                newElements.Add(element.CloneNode(true));
                continue;
```

```
            }

            var newParagraph = new Paragraph();
            foreach (var pEl in paragraph.Elements())
            {
               if (currentByte >= secret.Length)
               {
                  newParagraph.AppendChild(pEl.CloneNode(true));
                  continue;
               }

               if (pEl is not Run run)
               {
                  newParagraph.AppendChild(pEl.CloneNode(true));
                  continue;
               }

               for (int left = 0, right = 0; right < run.InnerText.Length; right++)
               {
                  var c = run.InnerText[right];
                  if (c == ' ' || c == '\t' || c == '\r' || c == '\n' || c == '\f' || c == '\v' || c ==
160)
                  {
                     var text = run.InnerText[left..right];
                     var textRun = new Run(new Text(text));
                     //textRun.RunProperties = new
RunProperties(run.RunProperties?.ChildElements ?? new());
                     textRun.RunProperties = new RunProperties();
```

```
                foreach (var childElement in run.RunProperties?.ChildElements ??
Enumerable.Empty<OpenXmlElement>())
                {

textRun.RunProperties.AppendChild(childElement.CloneNode(true));
                }

                newParagraph.AppendChild(textRun);



                var colorRed = currentByte < secret.Length ? secret[currentByte++]
: -1;
                var colorGreen = currentByte < secret.Length ?
secret[currentByte++] : -1;
                var colorBlue = currentByte < secret.Length ?
secret[currentByte++] : -1;
                if (colorRed == -1 && colorGreen == -1 && colorBlue == -1)
                {
                   newParagraph.AppendChild(new Run(new
Text(run.InnerText[right..])));
                   break;
                }

                var whitespace = new Text(c.ToString())
                {
                   Space = new
EnumValue<SpaceProcessingModeValues>(SpaceProcessingModeValues.Preserve)
                };
                var whitespaceRun = new Run(whitespace);
                whitespaceRun.RunProperties = new RunProperties();
```

```
                    foreach (var childElement in run.RunProperties?.ChildElements ??
Enumerable.Empty<OpenXmlElement>())
                    {

whitespaceRun.RunProperties.AppendChild(childElement.CloneNode(true));
                    }

                    colorRed = colorRed == -1 ? 0 : colorRed;
                    colorGreen = colorGreen == -1 ? 0 : colorGreen;
                    colorBlue = colorBlue == -1 ? 0 : colorBlue;

                    whitespaceRun.RunProperties.Color = new Color
                    { Val = $"{colorRed:X2}{colorGreen:X2}{colorBlue:X2}" };

                    newParagraph.AppendChild(whitespaceRun);

                    left = right + 1;
                }
            }
        }

        newElements.Add(newParagraph);
    }

    // Clear the body and add the new paragraphs
    body.RemoveAllChildren();
    foreach (var element in newElements)
    {
        body.AppendChild(element);
    }
```

```
    wordDoc.MainDocumentPart.Document.Save();
}

string Extract(string fullContainer)
{
    using var wordDoc = WordprocessingDocument.Open(fullContainer, false);
    var body = wordDoc.MainDocumentPart!.Document.Body;
    if (body == null)
    {
        Console.WriteLine("Document body is empty.");
        return "";
    }

    var hiddenBytes = new List<byte>();

    foreach (var paragraph in body.Elements<Paragraph>())
    {
        foreach (var run in paragraph.Elements<Run>())
        {
            var runProperties = run.RunProperties;
            if (runProperties?.Color != null)
            {
                string color = runProperties.Color.Val;
                if (color.Length == 6)
                {
                    try
                    {
                        byte r = Convert.ToByte(color.Substring(0, 2), 16);
                        byte g = Convert.ToByte(color.Substring(2, 2), 16);
```

```
                    byte b = Convert.ToByte(color.Substring(4, 2), 16);

                    if (r != 0 || g != 0 || b != 0)
                    {
                        hiddenBytes.Add(r);
                        hiddenBytes.Add(g);
                        hiddenBytes.Add(b);
                    }
                }
                catch (FormatException) { }
            }
        }
    }

    string extractedMessage =
Encoding.UTF8.GetString(hiddenBytes.ToArray());

    Console.WriteLine("Extracted message: " + extractedMessage);

    return extractedMessage;
  }
 }
}
```