Форма № 24

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет "Запорізька політехніка"

Факультет інформаційної безпеки та електронних комунікацій
(повне найменування інституту, факультету)

Кафедра інформаційної безпеки та наноелектроніки
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)
магістра
(ступінь вищої освіти)

на тему Алгоритм шифрування інформації в текстовому файлі
(назва теми)

Виконав(ла): студент(ка) 2 курсу, групи БКз-814м

Спеціальності 125 Кібербезпека та захист інформації
(код і найменування спеціальності)

Освітня програма (спеціалізація)

Безпека інформаційних і комунікаційних систем

ХАМЕТОВ А.А.
(ПРІЗВИЩЕ та ініціали)

Керівник КОЗІНА Г.Л.
(ПРІЗВИЩЕ та ініціали)

Рецензент МОРОЗ Г.В.
(ПРІЗВИЩЕ та ініціали)

2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет інформаційної безпеки та електронних комунікацій
Кафедра інформаційної безпеки та наноелектроніки
Ступінь вищої освіти магістр
Спеціальність 125 Кібербезпека та захист інформації
(код і найменування)
Освітня програма (спеціалізація) Безпека інформаційних і комунікаційних систем
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
Завідувач кафедри ІБтаН
Андрій КОРОТУН
«____» _____2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ХАМЕТОВА Андрія Алімовича
(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Алгоритм шифрування інформації в текстовому файлі Algorithm for encrypting information in a text file.
керівник проєкту (роботи) канд. фіз.-мат. наук., доцент КОЗІНА Галина Леонідівна
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)
затверджені наказом закладу вищої освіти від «26» листопада 2025 року № 530
2. Строк подання студентом проєкту (роботи) 22.12.2025
3. Вихідні дані до проєкту (роботи) Вихідними даними для виконання магістерської роботи є наукові публікації та монографії з теорії криптографії, симетричних алгоритмів шифрування та стеганографії, державні й міжнародні стандарти у сфері захисту інформації, приклади існуючих алгоритмів шифрування текстових даних, методи криптоаналізу та оцінювання криптостійкості.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) У роботі розглянуті такі питання: аналіз сучасного стану криптографії та методів захисту текстової інформації; дослідження симетричних алгоритмів шифрування та принципів їх роботи; огляд існуючих стеганографічних методів для текстових файлів; формулювання вимог до нового алгоритму шифрування інформації; побудова математичної моделі та структурної схеми алгоритму; програмна реалізація процесів шифрування і дешифрування; проведення експериментальних досліджень працездатності та криптостійкості; порівняльний аналіз із відомими аналогами; формування висновків і практичних рекомендацій щодо застосування розробленого алгоритму.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Презентація доповіді (в MS PowerPoint), 13 слайдів.

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3	КОЗІНА Г.Л., доцент кафедри ІбтаН	04.09.2025	19.12.2025
Нормоконтроль	КОРОЛЬКОВ Р.Ю., доцент кафедри ІбтаН		22.12.2025

7. Дата видачі завдання «04» вересня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1.	Аналіз літературних джерел за тематикою дослідження.	04.09.25 – 20.09.25	Виконано
2.	Формулювання мети, завдань, об’єкта та предмета дослідження, розробка структури роботи.	21.09.25 – 30.09.25	Виконано
3.	Розроблення моделі дослідження та структури програмної реалізації алгоритму.	01.10.25 – 15.10.25	Виконано
4.	Реалізація алгоритму шифрування та дешифрування текстових файлів.	16.10.25 – 10.11.25	Виконано
5.	Проведення експериментів і первинна обробка результатів.	11.11.25 – 01.12.25	Виконано
6.	Аналіз результатів, формування висновків та рекомендацій.	02.12.25 – 10.12.25	Виконано
7.	Оформлення пояснювальної записки, підготовка доповіді та презентації до захисту.	11.12.25 – 22.12.25	Виконано

Студент(ка) _____ Андрій ХАМЕТОВ
(підпис) (Ім’я ПРІЗВИЩЕ)

Керівник проєкту (роботи) _____ Галина КОЗІНА
(підпис) (Ім’я ПРІЗВИЩЕ)

АНОТАЦІЯ

Пояснювальна записка до магістерської роботи: 80 с., 4 табл., 12 рис., 8 джерел.

ТЕКСТОВИЙ ФАЙЛ, АЛГОРИТМ, ШИФРУВАННЯ, СИМЕТРИЧНА КРИПТОГРАФІЯ, СТЕГАНОГРАФІЯ, ЗАХИСТ ІНФОРМАЦІЇ.

Об’єктом дослідження є процес шифрування інформації в текстових файлах.

Предметом дослідження є методи та алгоритми симетричного шифрування і стеганографічного приховування даних у текстових контейнерах.

Метою магістерської роботи є розроблення та дослідження алгоритму шифрування інформації в текстовому файлі з оцінюванням його криптостійкості та практичної ефективності.

У роботі виконано аналіз сучасних підходів до захисту текстової інформації, зокрема симетричних алгоритмів шифрування, принципів криптоаналізу та методів стеганографії. Розглянуто основні класи симетричних шифрів, включаючи шифри перестановки, моно- та поліалфавітні підстановки, композиційні та ітераційні шифри, а також мережі Файстеля. Проаналізовано методи криптоаналізу та показано їхній вплив на оцінювання стійкості криптографічних систем.

Запропоновано новий алгоритм шифрування інформації у текстових файлах, який поєднує елементи симетричного криптографічного перетворення та стеганографічного маскування даних. Розроблено математичну модель алгоритму, структурну схему його функціонування та програмну реалізацію процесів шифрування і дешифрування.

У ході експериментальних досліджень здійснено перевірку працездатності алгоритму, оцінено його швидкодію, обсяг прихованих даних та стійкість до основних методів криптоаналізу. Отримані результати порівняно з показниками відомих програмних аналогів, що дозволило визначити переваги та обмеження запропонованого підходу.

ABSTRACT

Explanatory note to the Master's thesis: 80 pages, 4 tables, 12 figures, 8 references.

TEXT FILE, ALGORITHM, ENCRYPTION, SYMMETRIC CRYPTOGRAPHY, STEGANOGRAPHY, INFORMATION SECURITY.

The object of the study is the process of encrypting information in text files.

The subject of the study is the methods and algorithms of symmetric encryption and steganographic data hiding in text containers.

The purpose of the master's thesis is to develop and investigate an algorithm for encrypting information in a text file with an assessment of its cryptographic strength and practical efficiency.

The paper analyzes modern approaches to protecting textual information, including symmetric encryption algorithms, principles of cryptanalysis, and steganographic methods. The main classes of symmetric ciphers are considered, including transposition ciphers, monoalphabetic and polyalphabetic substitutions, composite and iterative ciphers, as well as Feistel networks. Cryptanalysis methods are analyzed and their influence on evaluating the security of cryptographic systems is shown.

A new algorithm for encrypting information in text files is proposed, which combines elements of symmetric cryptographic transformation and steganographic data masking. A mathematical model of the algorithm, its structural scheme, and a software implementation of the encryption and decryption processes are developed.

During the experimental studies, the operability of the algorithm was verified, its performance, the amount of hidden data, and its resistance to the main cryptanalysis methods were evaluated. The obtained results were compared with the indicators of known software analogues, which made it possible to determine the advantages and limitations of the proposed approach.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AES – Advanced Encryption Standard – алгоритм симетричного блочного шифрування

ASCII – American Standard Code for Information Interchange – стандарт кодування символів

BER – Bit Error Rate – імовірність бітової помилки

CR – Carriage Return – символ повернення каретки

CR/LF – Carriage Return / Line Feed – маркери кінця рядка

CSPRNG – Cryptographically Secure Pseudorandom Number Generator – криптографічно стійкий генератор псевдовипадкових чисел

CTR – Counter Mode – режим роботи блокового шифру

DES – Data Encryption Standard – стандарт симетричного шифрування

XOR – Exclusive OR – логічна операція «виключне АБО»

ВСТУП

Актуальність теми дослідження зумовлена стрімким зростанням обсягів цифрової інформації та підвищенням вимог до її захисту в умовах розвитку сучасних інформаційно-комунікаційних систем. У процесі передавання та зберігання даних усе частіше використовуються відкриті або потенційно контрольовані канали зв’язку, що підвищує ризики несанкціонованого доступу, перехоплення та аналізу інформаційних повідомлень. У таких умовах забезпечення конфіденційності інформації потребує застосування не лише класичних криптографічних методів, а й додаткових підходів, здатних приховувати сам факт передавання захищених даних.

Одним із таких підходів є стеганографія — сукупність методів прихованого передавання інформаційних повідомлень, що дозволяє маскувати наявність секретних даних у звичайному цифровому контенті. На відміну від криптографії, яка зосереджується на утаємниченні змісту повідомлення, стеганографія спрямована на приховування самого факту його існування. Поєднання цих методів забезпечує підвищений рівень захисту інформаційного обміну та ускладнює виявлення захищених повідомлень засобами пасивного аналізу.

Стеганографія має тривалу історію розвитку. За свідченням античних джерел, перші методи прихованого передавання інформації застосовувалися ще в Давній Греції. Одним із відомих прикладів є спосіб, за допомогою якого повідомлення передавали шляхом нанесення тексту на поголену голову посланця. Перше наукове згадування терміна «стеганографія» датується 1499 роком, коли Йоганн Трітеміус опублікував працю «Steganographia», присвячену методам прихованого обміну інформацією.

У сучасних умовах стеганографічні методи широко застосовуються в цифровому середовищі, зокрема для приховування даних у текстових, графічних, аудіо- та відеофайлах. Разом із тим зростає і кількість загроз, пов’язаних із

використанням стеганографії в шкідливих цілях, що ускладнює виявлення прихованих інформаційних потоків та потребує розроблення більш ефективних і стійких алгоритмів захисту інформації.

З огляду на зазначене, актуальним є дослідження та розроблення алгоритмів утаємничування інформації в текстових файлах, які поєднують переваги симетричного шифрування та стеганографічного маскування даних. Такі алгоритми дозволяють забезпечити підвищений рівень конфіденційності інформаційного обміну без використання спеціалізованих каналів зв’язку.

Метою даної дипломної роботи є розроблення та дослідження алгоритму шифрування інформації в текстовому файлі з використанням стеганографічних методів, а також оцінювання його криптостійкості та практичної ефективності.

Для досягнення поставленої мети в роботі передбачається аналіз сучасних методів криптографії та стеганографії, розроблення структури та програмної реалізації алгоритму приховування інформації, проведення експериментальних досліджень і формування практичних рекомендацій щодо застосування запропонованого підходу.

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Криптографія

Криптографія – це наука про інформаційну безпеку.

Криптографія – це галузь знань, що вивчає і розробляє криптографічні алгоритми перетворення даних.

У цьому сторіччі здійснюється перехід від класичної криптографії до сучасної асиметричної криптографія.

Класична криптографія ґрунтується на шифруванні – перетворенні даних на таку форму їхнього подавання, що робить неможливим розуміння змісту переробленого інформаційного повідомлення, не володіючи криптографічними ключами.

Криптографічний ключ – це деяка додаткова інформація, необхідна для відновлення початкового інформаційного повідомлення.

Накладання криптографічного ключа на інформаційне повідомлення забезпечує збереження інформації у таємниці її змісту від того, кому її не було призначено.

Основними задачами забезпечення захисту інформації є забезпечення:

– конфіденційності інформації;

– захисту від доступу до інформаційних повідомлень;

– захисту від копіювання інформаційних повідомлень;

– доступу до інформаційних повідомлень саме тим, кому він визначений за призначенням;

– захисту інформації від спотворення;

– збереження цілісності інформаційних повідомлень у такому вигляді, щоб респондент мав змогу визначити чи мали місце спотворення повідомлень в процесі їхнього передавання;

– використання гешів;

– використання засобів імітаційного захисту;

– розпізнавання справжності отриманих інформаційних повідомлень;

– розпізнавання справжності чинних користувачів;

– розпізнавання справжності авторства.

За цього, під розпізнаванням справжності отриманих інформаційних повідомлень розуміють можливість:

– здійснення їхнього обліку у каналі прийомо-передачі;

– додавання особливих ознак, що за аналізу авторства інформаційного повідомлення стануть визначальними;

– накладання особистого підпису на документи, що за аналізу авторства документу стануть визначальними.

За цього, під розпізнаванням справжності авторства розуміють і випадки необхідності доведення факту надсилання інформаційного повідомлення у випадку, коли джерело інформації заперечує свою причетність до процесів:

– створення;

– виготовлення;

– поширення;

– повідомлення про створення;

– повідомлення про надсилання інформаційного повідомлення.

Усі вище перелічені задачі зважуються шляхом застосування:

– засобів;

– методів криптографічного захисту даних.

Не зважаючи на те, що криптографію визискують вже кілька тисячоріч, її лише півстоліття як визнано галуззю наукової діяльності. Цей короткий відрізок часу є періодом інтенсивного розвитку:

-– відкритих досліджень;

– закритих розробок у різних галузях застосування математики в криптографії.

Провідною організацією міжнародного рівня у галузі криптографії є ISO.

Основним видом діяльності ISO є розробка відкритих стандартів.

1.2 Становлення сучасної криптографії

Відповідно до [1-3], добу сучасної криптографії було започатковано відомим американським науковцем Шеноном Клодом Елвудом, що працював на посаді професора Масачусетського технологічного інституту (рис.1.1).

Рисунок 1.1 – Масачусетський технологічний інститут

Окрім того, Шенон мав родинні зв’язки з одним із найвизначніших винахідників в історії людства – Томасом Едісоном. Імовірно, саме це оточення сприяло формуванню інтересу молодого Клода до технічної творчості, адже з раннього віку він захоплювався створенням різноманітних механізмів і пристроїв.

Винаходи, що зберігалися у його домашній колекції, вирізнялися не меншою оригінальністю, ніж розробки самого Едісона, якими людство користується й нині. Клод Елвуд Шенон загалом був надзвичайно багатогранною особистістю. Зокрема, він палко цікавився шахами й, навіть, створив механічний пристрій для гри в шахи.

Це було задовго до появи суперкомп’ютера Deep Blue, який компанія IBM побудувала для історичного матчу з Гарі Каспаровим. У 1965 році, перебуваючи з візитом у Радянському Союзі, Шенон навіть зіграв шахову партію з чемпіоном світу Михайлом Ботвінніком. Хоча він поступився на 42-му ході, його гра продемонструвала високий рівень майстерності.

Поєднання різноманітних талантів дозволяло Шенону по-особливому сприймати звичні речі, знаходячи нестандартні підходи до складних наукових проблем. Під час вручення Нобелівської премії його промова відзначалася стриманістю та гідністю, властивими видатному вченому. Він зауважив, що наукові досягнення в одній галузі можуть бути корисними й в інших сферах знань. Саме така здатність до міждисциплінарного мислення й зумовила його інтерес до криптографії, якою вона була в першій половині XX століття.

Оскільки криптографія значною мірою розвивалася в межах військових технологій, більшість ключових досягнень Шенона в цій галузі та в теорії інформації тісно пов’язані з військовими структурами. Відомо, що з 1941 року і аж до 1972 року він співпрацював не лише з дослідницькою лабораторією Bell Laboratories, а й з військовими відомствами США. Під час Другої світової війни Шенон разом із командою інженерів працював над створенням різних оборонних систем, зокрема радарів і засобів протиповітряної оборони для авіації.

Одним із результатів його діяльності стала система ідентифікації "свій-чужий", яка застосовувалася в американських військово-повітряних силах для розпізнавання нейтральних і союзних об’єктів та відокремлення їх від ворожих. Радарні системи, розроблені за його участі, протягом тривалого часу активно використовувалися розвідками різних країн.

Найціннішим підсумком співпраці між Шеноном і Управлінням стратегічних служб США стала поява праці "Математична теорія зв’язку", опублікованої у 1948 році в журналі The Bell System Technical Journal. Саме в ній були закладені основи теорії інформації та теорії зв’язку, які згодом стали фундаментом для численних наукових відкриттів. У цій роботі Шенон запропонував модель каналу зв’язку, що

включає джерело повідомлень, приймач і джерело шуму, розглядаючи їх як ймовірнісні об’єкти (рис. 1.2) [1-3].

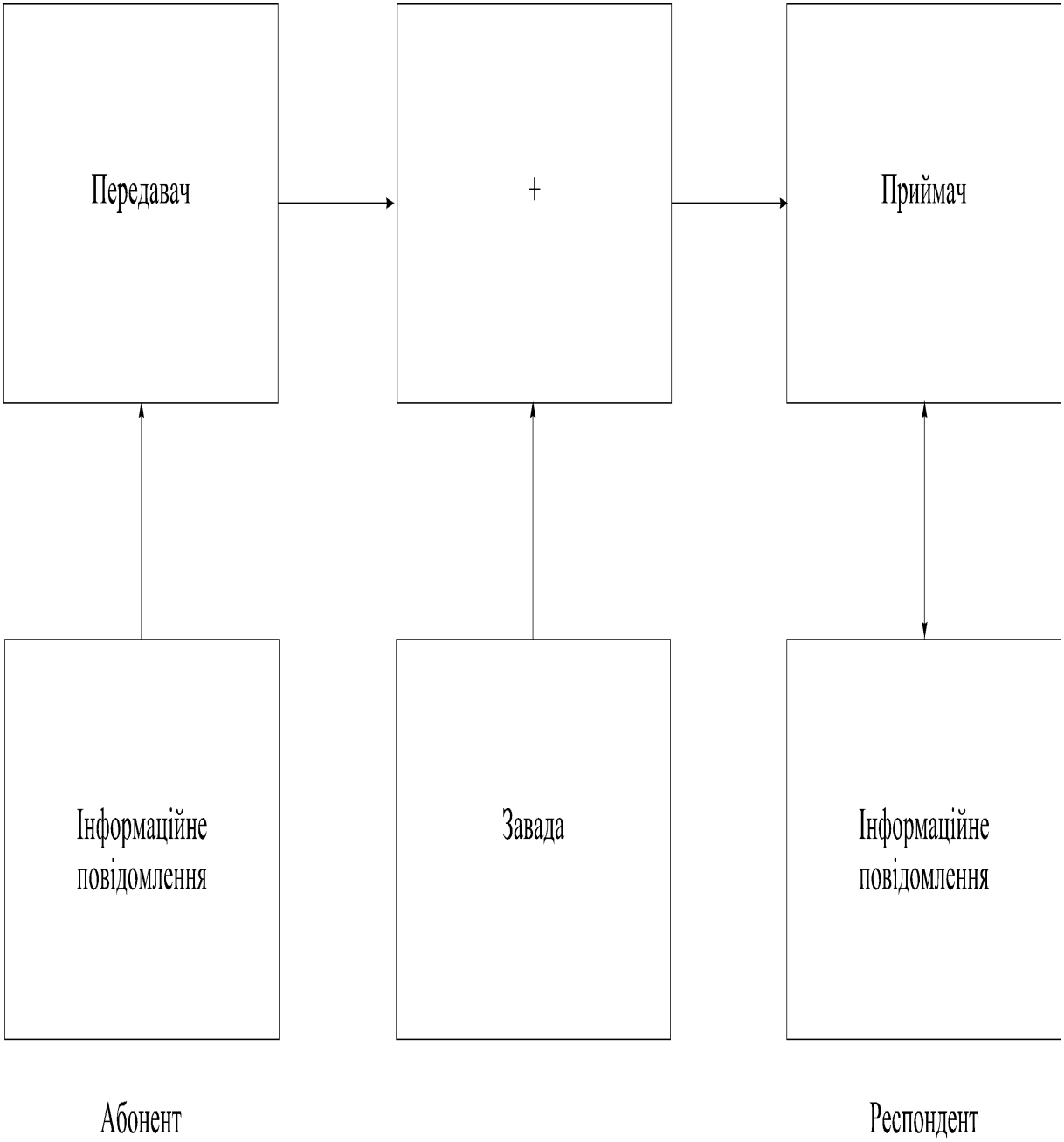


Рисунок 1.2 – Канал зв’язку

Шенон трактував джерело повідомлень як випадковий процес, який генерує та передає повідомлення з наперед визначеними ймовірностями. Для цього він використав класичний апарат теорії ймовірностей, розглядаючи повідомлення як дискретні елементи. У його інтерпретації кожне повідомлення являло собою неподільний інформаційний блок, що значно спрощувало математичний опис процесів передавання даних і дозволяло формалізувати їх у вигляді ймовірнісних співвідношень.

Подальший розвиток ідей, викладених у "Математичній теорії зв’язку", привів Шенона до формування фундаментальних положень теорії інформації. У межах цієї теорії він увів поняття кількісної міри інформації, що дозволило формалізувати процеси передавання та обробки повідомлень. Такий підхід зробив можливим порівняння різних джерел інформації з точки зору їх інформативності та ступеня невизначеності.

Ключовим поняттям у теорії інформації стала ентропія, яку Шенон запропонував як міру невизначеності випадкової величини. Для її обчислення він використав логарифмічну функцію, що дало змогу отримати зручний та узагальнений математичний апарат. Застосування двійкового логарифма було обумовлене особливостями двійкових систем числення, які лежать в основі роботи обчислювальної техніки.

Об’єднавши поняття ентропії та інформативності, Шенон отримав можливість визначати надмірність тексту. Для цього він проводив експерименти, у межах яких намагався передбачати наступні символи в текстах різними мовами. Аналізуючи результати таких експериментів, він встановив середню кількість інформації, що припадає на один символ, а також визначив статистичні властивості мовних повідомлень.

У своїх дослідженнях Шенон активно використовував результати робіт Маркова Андрія Андрійовича, зокрема теорію марковських процесів. Застосування цього підходу дозволило моделювати тексти як послідовності випадкових величин, залежних від попередніх станів. Такий статистичний опис став важливим

інструментом у подальшому розвитку методів аналізу відкритих і зашифрованих повідомлень.

Наступним етапом наукової діяльності Шенона стало вивчення таємних систем зв’язку. Він адаптував загальну модель каналу зв’язку до задач криптографії, замінивши джерело шуму на криптоаналітика, який намагається відновити зміст зашифрованого повідомлення (рис.1.3) [1-3]. У цій моделі вважалося, що алгоритм шифрування є відомим, а безпека системи забезпечується лише таємністю ключа.

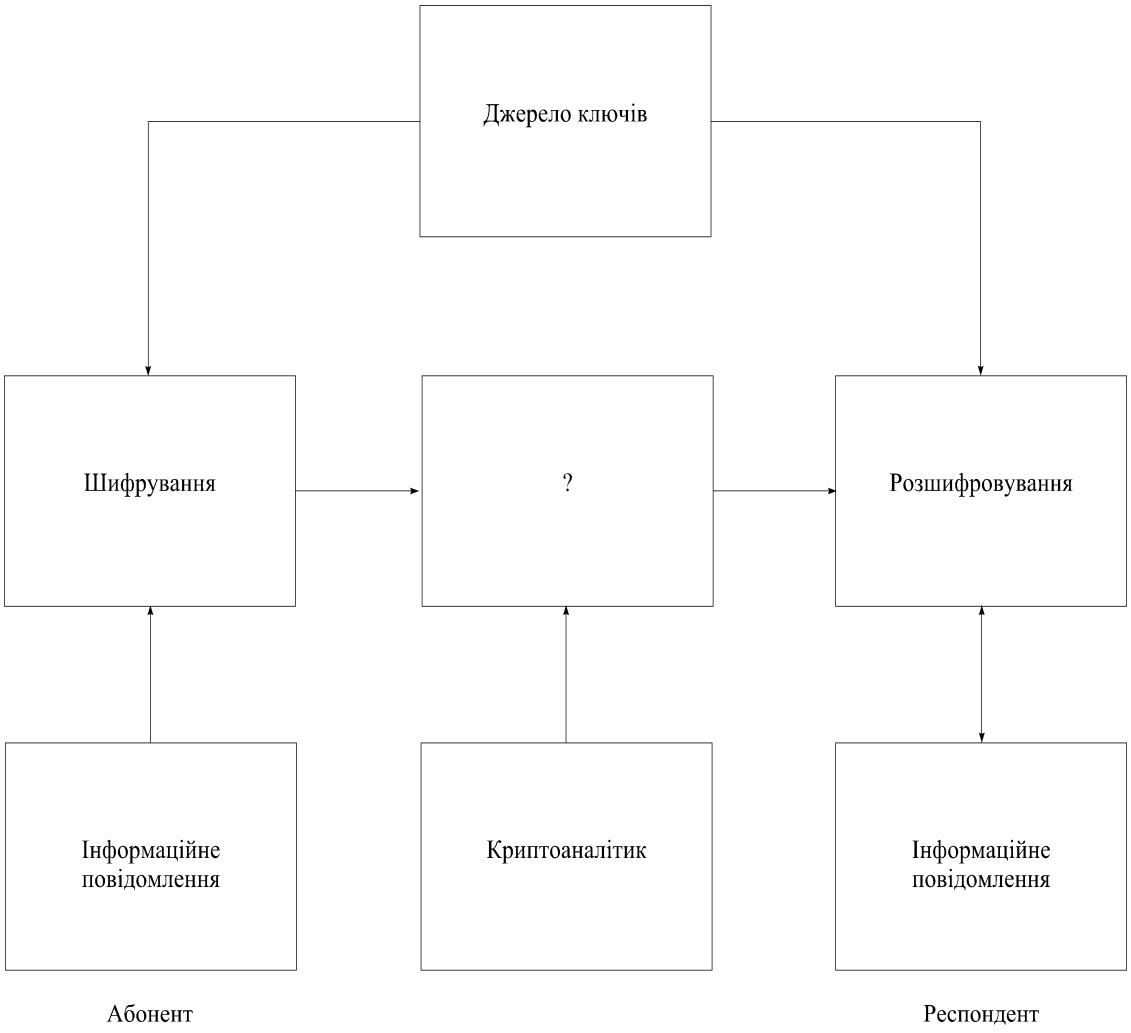


Рисунок 1.3 – Канал таємного зв'язку

Дотримуючись принципу Керкгоффса, Шенон наголошував, що надійність криптографічної системи не мусить залежати від таємності алгоритму. Він порівнював таку систему із замком, конструкція якого є відкритою, але комбінація залишається прихованою і може бути змінена у разі її компрометації. Цей підхід став основоположним для побудови більшості сучасних криптографічних алгоритмів.

У рамках теорії таємних систем Шенон також сформулював вимоги до ідеальної криптосистеми та ввів поняття абсолютної стійкості. Він довів, що для досягнення такої стійкості ентропія ключа повинна бути не меншою за ентропію відкритого тексту. Якщо ця умова не виконується, криптографічна система втрачає властивість повної захищеності.

1.3 Види симетричних шифрів

Симетричними називають такі криптографічні алгоритми, у яких для виконання операцій шифрування та зворотного перетворення використовується один і той самий таємний ключ. Залежно від способу обробки відкритого тексту всі симетричні шифри поділяють на кілька основних класів, зокрема шифри перестановки, моноалфавітні та поліалфавітні заміни, композиційні й ітераційні шифри, а також криптографічні системи, побудовані за схемою мережі Файстеля.

Шифри перестановки базуються на зміні порядку символів у повідомленні відповідно до певного правила, що задається ключем. При цьому самі символи залишаються незмінними, а захист інформації досягається виключно за рахунок приховування їх початкової послідовності. Такі шифри є одними з найдавніших і найпростіших з точки зору реалізації.

Моноалфавітні шифри заміни передбачають використання фіксованої відповідності між символами відкритого та зашифрованого текстів. У межах

такого підходу кожному символу початкового алфавіту ставиться у відповідність інший символ, причому ця відповідність зберігається протягом усього процесу шифрування.

Поліалфавітні шифри заміни використовують декілька алфавітів, які застосовуються почергово або залежно від позиції символу в повідомленні. У таких системах перетворення визначається не лише таємним ключем, а й номером символу у тексті. Якщо алфавіти відкритого та зашифрованого текстів збігаються, то відповідну поліалфавітну заміну прийнято називати підстановкою.

Композиційні шифри будуються шляхом послідовного застосування кількох простіших криптографічних перетворень. Ключ композиційного шифру зазвичай складається з набору циклових ключів, кожен з яких використовується на окремому етапі шифрування. Якщо всі цикли мають однакову структуру, такий шифр називають ітераційним.

Окремий клас композиційних шифрів утворюють мережі Файстеля. У таких системах відкритий текст поділяється на дві рівні частини, над якими виконуються послідовні перетворення з використанням циклових функцій і операцій обміну. За умови достатньої кількості раундів і коректного вибору ключів мережі Файстеля забезпечують високий рівень криптографічної стійкості.

Симетричні криптографічні алгоритми також класифікують за способом обробки даних на блокові та потокові. У блокових шифрах інформація обробляється фіксованими за розміром блоками, причому кожен блок шифрується незалежно від інших. Потокові шифри, навпаки, працюють із безперервним потоком даних, а перетворення кожного символу або біта може залежати від його позиції або попередніх елементів. У багатьох випадках потокові шифри реалізуються на основі гамування, де рівень безпеки визначається властивостями псевдовипадкової послідовності.

1.4 Принципи криптоаналізу

Криптоаналізом називають сукупність методів і підходів, спрямованих на дослідження криптографічних систем з метою відновлення відкритого тексту або визначення таємного ключа без його прямого знання. Процес здійснення такого дослідження прийнято називати атакою на шифр. Успішним зламом вважається знаходження такої слабкої ланки в системі, яка дозволяє виконати атаку з меншою обчислювальною складністю, ніж повний перебір усіх можливих ключів.

Фундаментальним припущенням криптоаналізу є правило Керкгоффса, відповідно до якого алгоритм шифрування вважається повністю відомим потенційному противнику. За таких умов стійкість криптографічної системи визначається виключно таємністю ключа, а не прихованістю структури алгоритму. Це припущення лежить в основі більшості сучасних моделей аналізу безпеки.

Залежно від обсягу інформації, якою володіє криптоаналітик, розрізняють кілька основних типів атак. До них належать атаки зі знанням лише шифротексту, атаки з відомим відкритим текстом, атаки з обраним відкритим текстом, атаки з обраним шифротекстом, а також адаптивні атаки, у яких стратегія аналізу змінюється в процесі отримання нових даних.

Для формального опису стійкості криптосистем використовують поняття апріорної та апостеріорної невизначеності. Апріорна невизначеність характеризує ступінь непередбачуваності відкритого тексту до отримання шифротексту і визначається його ентропією. Апостеріорна невизначеність, у свою чергу, відображає залишкову невизначеність після того, як шифротекст уже відомий криптоаналітику.

Різниця між апріорною та апостеріорною невизначеністю показує кількість інформації, яку можна отримати з аналізу шифротексту. Криптографічні системи, для яких ці дві величини збігаються, називають

абсолютно стійкими. Клод Шенон довів, що необхідною умовою досягнення абсолютної стійкості є те, щоб ентропія таємного ключа була не меншою за ентропію відкритого тексту.

У випадку, коли зазначена умова не виконується, криптографічна система вважається недосконалою, оскільки з часом можливе накопичення інформації, достатньої для її зламу. Шенон також увів поняття надмірності мови та показав, що відстань одиничності шифру безпосередньо залежить від надмірності відкритого тексту і довжини ключа. За відсутності надмірності, криптоаналіз, заснований лише на аналізі шифротексту, стає неможливим навіть за умови необмежених обчислювальних ресурсів.

1.5 Стійкі до стегоаналізу методи реалізації стеганографії

Безпека телекомунікацій завжди є актуальною проблемою, особливо у сучасну цифрову добу. Зростання кількості кібератак і поява нових методів зламу даних є прикрою тенденцією. Криптографія – це техніка шифрування інформації з метою її захисту від несанкціонованого доступу. Вона є способом забезпечення обміну особистими повідомленнями. Але вона залишається вразливою для атак, залишаючи небезпеку [4].

Пліч опліч з криптографією іде стеганографія – техніка приховання інформації усередині довільного змісту (рис. 1.4).

Рисунок 1.4 – Стеганографія

Цим можуть бути зображення, текстові, аудіо- й відео-файли. На відміну від криптографії, за використання цього методу приховується сам факт наявності того, що щось було заховано. Така здатність забезпечувати стискання й компактніше збереження даних робить стеганографію надзвичайно привабливим методом.

Тобто, стеганографію можна вважати більш ефективним способом обміну повідомленнями. Однак ризик розкриття даних не зникає.

Фахівці завжди мріють розробили алгоритм, а користувачі – нарешті його отримати, для незламної реалізації стеганографії.

Метою незламної реалізації стеганографії є приховування інформаційного повідомлення настільки ефективно, що виявлення факту її існування стає неможливим [5].

Незламна реалізація стеганографії мусить створити сферу безпечної комунікації, дозволивши найрізноманітнішим верствам населення вести вільне спілкування і захищати свою інформацію.

Незламна реалізація стеганографії, звісно, мусить ґрунтуватися на останніх досягненнях сучасної теорії інформації. Новий алгоритм мусить бути

здатний зашифровувати інформаційні повідомлення у довільний зміст, не змінюючи його вмісту. Тоді між статистичним розподілом даних загальної інформації й розподілом зашифрованої інформації не буде ніякої декореляції. Це зробить алгоритм набагато більш безпечним у порівнянні з відомими методами стеганографії.

Наявні методи стеганографії у різному ступені змінюють розподіл загального змісту, що уможливлює знаходження факту приховування в ньому даних.

Незламна реалізація стеганографії мусить забезпечувати збільшення ефективності шифрування з метою забезпечення можливості приховування більших обсягів інформації. І тоді вона стане здатною змінити способи обміну інформацією навіть у відкритих мережах, дозволивши вести більш вільне спілкування і захищати свою особисту інформацію.

Поява незламної реалізації стеганографії стане значним кроком у напрямку приватного спілкування.

1.6 Сфера та призначення застосування стеганографічних методів

Розроблений у даній магістерській роботі алгоритм утаємничування інформації в текстових файлах призначений для забезпечення прихованого та захищеного обміну даними в умовах відкритих або потенційно контрольованих каналів зв’язку. Основною особливістю запропонованого підходу є поєднання криптографічних перетворень із методами стеганографічного маскування [6], що дозволяє не лише ускладнити доступ до змісту повідомлення, а й приховати сам факт його існування.

Алгоритм орієнтований на роботу з текстовими контейнерами у стандартних форматах і не потребує використання спеціалізованих програмних або апаратних засобів. Це робить можливим його застосування у широкому

колі інформаційних систем, включаючи персональні комп’ютери, мобільні пристрої та мережеві сервіси. Простота програмної реалізації та відсутність жорстких вимог до обчислювальних ресурсів дозволяють інтегрувати алгоритм у прикладні програмні продукти без істотного зниження їхньої продуктивності.

Призначення алгоритму також полягає у підвищенні рівня конфіденційності інформаційного обміну в умовах, коли застосування класичних криптографічних методів може привертати небажану увагу або бути обмеженим організаційними чи технічними чинниками. Використання стеганографічних принципів дає змогу маскувати передавання зашифрованих даних під звичайний текстовий контент, що істотно ускладнює їх виявлення засобами пасивного аналізу.

Наведемо деякі приклади з можливих областей застосування незламної реалізація стеганографії:

1.6.1 Використання стеганографічних методів у соціальних мережах

Соціальні мережі (рис. 1.5) усе більше використовуються не лише для особистого спілкування, а й для розв’язання професійних, організаційних та інформаційно-аналітичних завдань. Через значні обсяги передаваних даних і масовий характер комунікацій такі платформи становлять підвищений інтерес для систем моніторингу, аналізу трафіку та несанкціонованого збору інформації.

Рисунок 1.5 – Соціальні мережі

Особливістю більшості сучасних соціальних мереж є централізована архітектура зберігання та обробки повідомлень, що передбачає контроль з боку адміністраторів сервісів, а в окремих випадках – державних або корпоративних структур. Навіть за умови використання стандартних засобів криптографічного захисту, таких як транспортне шифрування, метадані повідомлень (час передавання, обсяг, частота обміну) залишаються доступними для аналізу, що може призвести до компрометації інформаційної взаємодії.

У цьому контексті застосування алгоритму незламної реалізації стеганографії дозволяє підвищити рівень конфіденційності спілкування за рахунок маскування зашифрованих повідомлень під звичайний текстовий контент. Передавання прихованої інформації у вигляді звичайних текстових повідомлень або публікацій не викликає підозр з боку систем автоматизованого контролю та не потребує використання спеціалізованих каналів зв’язку.

Запропонований алгоритм може бути використаний для прихованого обміну повідомленнями у приватних чатах, коментарях, описах публікацій або інших текстових елементах соціальних платформ. При цьому зовнішній вигляд і семантична цілісність тексту зберігаються, що ускладнює виявлення факту приховування інформації як автоматизованими засобами аналізу, так і

пересічними користувачами.

Таким чином, використання розробленого алгоритму у соціальних мережах створює додатковий рівень захисту інформаційної взаємодії, поєднуючи переваги криптографічного шифрування з можливостями стеганографічного маскування, що є особливо актуальним в умовах зростання контролю та аналізу цифрових комунікацій.

1.6.2 Оцінювання ефективності стеганографічних методів

Ефективність застосування незламної реалізації стеганографії визначається сукупністю показників, до яких належать пропускна здатність каналу прихованого передавання, обчислювальна складність алгоритму, обсяг модифікацій контейнера та рівень стійкості до виявлення. Запропонований у роботі алгоритм орієнтований на досягнення балансу між цими параметрами, що забезпечує його практичну доцільність для використання в реальних інформаційних системах.

Однією з ключових переваг алгоритму є раціональне використання надмірності текстового контейнера. Завдяки поєднанню симетричного шифрування з методами прихованого кодування бітової інформації досягається можливість вбудовування повідомлень без істотного збільшення розміру файла та без помітного спотворення його структури. Це дозволяє ефективно приховувати інформацію навіть у відносно невеликих текстових об’ємах.

З точки зору обчислювальних витрат алгоритм не потребує виконання складних математичних операцій або використання ресурсоємних криптографічних примітивів. Основні перетворення реалізуються за допомогою операцій побітової обробки, роботи з символами та псевдовипадковими послідовностями, що забезпечує високу швидкодію та можливість використання алгоритму на пристроях з обмеженими обчислювальними

ресурсами.

Важливим аспектом ефективності є також стійкість алгоритму до статистичного та структурного аналізу. Мінімальні зміни, які вносяться до текстового контейнера, не призводять до істотної зміни розподілу символів або форматування, що ускладнює виявлення прихованої інформації засобами автоматизованого аналізу. Таким чином, ефективність алгоритму проявляється не лише у швидкості його роботи, а й у здатності тривалий час залишатися непомітним у відкритих каналах передавання.

Отже, запропонований алгоритм характеризується достатнім рівнем ефективності для практичного застосування, поєднуючи помірні обчислювальні витрати, прийнятну пропускну здатність та високий рівень прихованості інформаційного обміну, що відповідає сучасним вимогам до систем захисту текстової інформації.

1.6.3 Застосування стеганографічних методів у фаховій діяльності

У межах загальної фахової діяльності значна кількість спеціалістів має справу з обробкою, зберіганням та передаванням конфіденційної або службової інформації. До таких фахівців належать, зокрема, працівники правоохоронних органів, медичні працівники, журналісти, юристи, науковці, а також спеціалісти у сфері інформаційних технологій та кібербезпеки. У процесі виконання професійних обов’язків вони часто обмінюються даними, розголошення яких може призвести до порушення професійної таємниці, завдання матеріальної шкоди або негативних правових наслідків.

У сучасних умовах фахова комунікація дедалі частіше здійснюється з використанням електронних засобів зв’язку, включаючи електронну пошту, месенджери, хмарні сервіси та соціальні платформи. За таких умов інформація перебуває під постійною загрозою перехоплення, аналізу або

несанкціонованого доступу. Навіть застосування стандартних криптографічних механізмів не завжди забезпечує належний рівень захисту, оскільки сам факт передавання зашифрованих повідомлень може привертати увагу та ставати об’єктом додаткового аналізу.

Застосування алгоритму незламної реалізації стеганографії у фаховій діяльності дозволяє мінімізувати зазначені ризики шляхом приховування зашифрованої інформації у звичайному текстовому контенті. Такий підхід забезпечує додатковий рівень захисту, оскільки ускладнює не лише доступ до змісту повідомлення, а й виявлення самого факту його існування. Це є особливо важливим у ситуаціях, коли необхідно зберегти конфіденційність професійного спілкування без використання спеціалізованих або помітних засобів захисту.

Розроблений алгоритм може бути використаний для безпечного обміну службовими повідомленнями, передачі текстових звітів, коментарів, внутрішніх інструкцій та іншої інформації, що має обмежений доступ. Завдяки простоті програмної реалізації та невисоким вимогам до обчислювальних ресурсів його застосування не потребує зміни існуючих робочих процесів і може бути інтегроване у наявні інформаційні системи.

Таким чином, використання запропонованого алгоритму у загальній фаховій діяльності сприяє підвищенню рівня інформаційної безпеки, збереженню професійної таємниці та зниженню ризиків, пов’язаних із несанкціонованим розкриттям або аналізом службових даних у цифровому середовищі.

1.6.4 Застосування стеганографічних методів в умовах інформаційних обмежень

В умовах посиленого контролю інформаційних потоків та обмеження свободи комунікацій у окремих інформаційних середовищах особливого

значення набувають методи захисту приватного обміну даними. У таких умовах користувачі можуть стикатися з необхідністю забезпечення конфіденційності спілкування та збереження анонімності під час передавання інформації, що має суспільно значущий або чутливий характер.

Алгоритми прихованого передавання інформації можуть застосовуватися для захисту комунікацій у ситуаціях, коли використання відкритих криптографічних засобів є небажаним або обмеженим через ризик виявлення самого факту захищеного обміну даними. Стеганографічні методи дозволяють маскувати зашифровані повідомлення у звичайному текстовому контенті, що ускладнює їх ідентифікацію засобами автоматизованого моніторингу та аналізу.

Запропонований у роботі алгоритм може бути використаний для прихованого обміну текстовою інформацією в умовах підвищеного контролю цифрових каналів зв’язку. Його застосування дозволяє знизити ймовірність виявлення комунікації як такої, зберігаючи при цьому конфіденційність змісту переданих даних. Такий підхід є особливо актуальним для інформаційних середовищ із жорсткими обмеженнями на свободу обміну повідомленнями або з активним аналізом інформаційного трафіку.

Важливо зазначити, що використання подібних алгоритмів має здійснюватися з урахуванням чинного законодавства та етичних норм. У контексті даної роботи розглядається виключно технічний аспект забезпечення захищеної та прихованої комунікації як складової сучасних систем інформаційної безпеки.

Таким чином, застосування алгоритму незламної реалізації стеганографії в умовах протидії надмірному інформаційному контролю сприяє збереженню конфіденційності спілкування та підвищенню стійкості інформаційного обміну без порушення функціональної цілісності цифрових каналів зв’язку.

1.6.5 Особливості використання стеганографії в міжнародному інформаційному обміні

У межах закордонної діяльності питання захисту інформаційного обміну набувають особливої актуальності через використання різнорідних телекомунікаційних інфраструктур, відмінності у правових нормах та підвищений ризик перехоплення або аналізу передаваних даних. Особи, які здійснюють професійну, наукову, ділову або освітню діяльність за межами своєї країни, часто змушені користуватися відкритими або недостатньо захищеними каналами зв’язку.

Під час міжнародної комунікації текстові повідомлення можуть передаватися через мережі та сервіси, що перебувають під юрисдикцією інших держав або приватних компаній. У таких умовах існує ймовірність доступу до інформації з боку третіх осіб, а також застосування автоматизованих систем моніторингу та аналізу контенту. Навіть за наявності стандартних криптографічних механізмів захисту сам факт використання зашифрованих повідомлень може стати об’єктом підвищеної уваги.

Застосування алгоритму незламної реалізації стеганографії у закордонній діяльності дозволяє підвищити рівень захищеності інформаційного обміну шляхом приховування зашифрованих даних у звичайному текстовому контенті. Такий підхід зменшує ймовірність ідентифікації повідомлень як захищених або службових, що є важливим у середовищах із різними правилами інформаційної безпеки та контролю.

Запропонований алгоритм може використовуватися для безпечного обміну текстовою інформацією між партнерами з різних країн, під час участі у міжнародних проєктах, наукових дослідженнях, ділових перемовинах або освітніх програмах. Його універсальність і незалежність від конкретної платформи забезпечують можливість застосування в різних інформаційних

середовищах без потреби в адаптації до локальних стандартів.

Отже, використання розробленого алгоритму у закордонній діяльності сприяє підвищенню рівня конфіденційності та надійності міжнародного інформаційного обміну, знижує ризики несанкціонованого доступу до текстових даних та забезпечує додатковий рівень захисту у глобальному цифровому середовищі.

1.7 Особливості практичного застосування стеганографічних методів

Незважаючи на значні переваги, які надає незламна реалізація стеганографії для захисту інформаційного обміну, її практичне застосування має низку особливостей і обмежень, що повинні враховуватися під час використання алгоритму в реальних умовах. Ефективність прихованого передавання даних залежить не лише від властивостей самого алгоритму, а й від характеристик середовища, у якому здійснюється обробка та передавання текстових повідомлень.

Однією з ключових особливостей є залежність стеганографічних методів від цілісності текстового контейнера. Будь-яке автоматизоване редагування тексту, зокрема вирівнювання, заміна кількох пробілів одним, автоматична корекція пунктуації або конвертація файлів у інші формати, може призвести до часткової або повної втрати прихованої інформації [7]. Тому застосування алгоритму доцільне передусім у середовищах, де зберігається контроль над процесами обробки та передавання текстових даних.

Важливим чинником є також безпека кінцевих пристроїв, на яких виконується шифрування, приховування та витягування інформації. У разі компрометації пристрою шляхом перехоплення введення з клавіатури, запису екрана, акустичного зняття інформації або впровадження шкідливого програмного забезпечення ефективність будь-яких криптографічних чи

стеганографічних методів істотно знижується. Таким чином, застосування алгоритму повинно супроводжуватися дотриманням базових вимог інформаційної та кібербезпеки.

Ще однією особливістю є необхідність узгодження параметрів алгоритму між сторонами інформаційного обміну. Для коректного витягування прихованого повідомлення сторони повинні володіти інформацією щодо використовуваного ключа, методу приховування та порядку обробки контейнера. Порушення синхронізації або неправильне налаштування параметрів може унеможливити відновлення переданих даних.

Слід також враховувати, що застосування стеганографічних методів не замінює класичних засобів криптографічного захисту, а доповнює їх. Максимальний рівень безпеки досягається лише за умови поєднання шифрування змісту повідомлення з приховуванням факту його передавання. Такий підхід дозволяє знизити ризик як несанкціонованого доступу до інформації, так і її виявлення в процесі аналізу відкритих каналів зв’язку.

Отже, ефективне застосування запропонованого алгоритму можливе за умови комплексного підходу до захисту інформації, що враховує технічні, програмні та організаційні аспекти безпеки, а також особливості середовища функціонування текстових контейнерів.

1.8 Мета та завдання дослідження

Метою даної магістерської роботи є розроблення та дослідження алгоритму утаємничування інформації в текстових файлах, який поєднує методи симетричного шифрування з принципами стеганографічного приховування даних з метою підвищення рівня конфіденційності та прихованості інформаційного обміну в умовах відкритих каналів зв’язку.

Для досягнення поставленої мети в процесі виконання магістерської

роботи необхідно розв’язати такі основні завдання:

– здійснити аналіз сучасного стану криптографічних і стеганографічних методів захисту текстової інформації та визначити їхні переваги й обмеження;

– дослідити властивості симетричних алгоритмів шифрування та можливості їх використання у поєднанні зі стеганографічними методами приховування даних;

– обґрунтувати вимоги до алгоритму утаємничування інформації в текстових файлах з урахуванням ефективності, прихованості та стійкості до основних методів аналізу;

– розробити структурну схему та математичну модель запропонованого алгоритму;

– реалізувати програмні засоби для виконання процесів шифрування, приховування та витягування інформації з текстових контейнерів;

– провести експериментальні дослідження працездатності алгоритму, оцінити його швидкодію, обсяг прихованих даних та стійкість до основних методів крипто- та стегоаналізу;

– виконати порівняльний аналіз розробленого алгоритму з відомими програмними аналогами та сформулювати практичні рекомендації щодо його застосування.

Результати виконаного дослідження мають практичне та наукове значення і можуть бути використані під час розроблення програмних засобів захисту текстової інформації, у навчальному процесі при вивченні дисциплін з криптографії та інформаційної безпеки, а також як основа для подальших досліджень у сфері прихованого передавання даних.

2 СТЕГАНОГРАФІЧНІ МЕТОДИ ДЛЯ ТЕКСТОВИХ ФАЙЛІВ

2.1 Загальна характеристика методів

Загалом текстові методи стеганографії характеризуються порівняно низькою пропускною здатністю, що зумовлено обмеженою кількістю надлишкової інформації в тексті. На відміну від графічних або аудіоконтейнерів, текст не допускає значних змін без ризику порушення його читабельності або смислової цілісності. Саме тому для приховування навіть невеликих обсягів даних потрібні контейнери значного розміру.

Водночас простота реалізації та відсутність складних обчислювальних операцій роблять текстову стеганографію привабливою для навчальних і демонстраційних цілей. Описані методи легко реалізуються програмними засобами та не потребують спеціалізованого програмного забезпечення. Це дозволяє наочно продемонструвати базові принципи приховування та витягування інформації.

Однак на практиці застосування текстових стеганографічних методів суттєво обмежується впливом зовнішніх факторів. Автоматичне форматування, перевірка орфографії, копіювання тексту між різними редакторами або перетворення файлів у інші формати часто призводять до знищення прихованих даних. У результаті такі методи можуть бути використані лише в умовах повного контролю над середовищем обробки тексту.

Незважаючи на зазначені недоліки, текстова стеганографія залишається важливим напрямом досліджень у сфері захисту інформації. Її методи становлять теоретичний інтерес і можуть слугувати основою для розроблення більш стійких і адаптивних алгоритмів приховування даних, зокрема в поєднанні з іншими видами цифрових контейнерів.

2.2 Семантичні методи

Для маскування конфіденційної інформації в текстових повідомленнях (так звана лінгвістична стеганографія) застосовуються:

– особливості форматів подавання текстових даних;

– властивості надмірності письмової мови.

Текст у електронному вигляді є одним із найбільш складних об’єктів для приховування інформації з низки причин. На відміну від електронного текстового файла, його матеріальна копія (наприклад, надрукований документ) може розглядатися як високоструктуроване зображення, що робить її відносно придатною для різноманітних методів приховання даних. До таких методів належать:

– коригування відстаней між окремими символами (кернінг);

– незначні зміни параметрів форматування;

– коригування відстаней міжрядкових інтервалів тощо.

Подібна ситуація значною мірою пояснюється обмеженою кількістю надлишкової інформації в текстових файлах, особливо в порівнянні з графічними або аудіоданими. Якщо у зображеннях і звукових сигналах у більшості випадків можна внести зміни, непомітні для зору чи слуху, то навіть поява зайвого символу або знака пунктуації в тексті легко виявляється уважним читачем.

Процес приховування даних у тексті вимагає пошуку таких способів модифікації, які залишалися б непомітними для більшості користувачів. У роботі [8] найпоширеніші методи для приховування інформації до текстових файлів класифіковано на три групи:

– синтаксичні методи, що використовують особливості пунктуації;

– семантичні методи, алгоритми яких ґрунтуються на зміні слів відповідно до вибраного закону приховування даних;

– методи довільного інтервалу, алгоритми яких ґрунтуються на маніпуляціях з відстанями всередині тексту:

а) між словами;

б) між рядками.

Той факт, що вільний простір для вбудовування інформації обирається довільно, одночасно є як перевагою, так і недоліком з погляду прихованості даних. Людина, яка читає текст, може не помітити внесених змін, тоді як програмні засоби редагування здатні автоматично змінювати кількість і розташування пробілів, унаслідок чого приховані дані руйнуються.

Синтаксичні методи приховування інформації в текстових повідомленнях ґрунтуються на використанні особливостей розділових знаків і правил пунктуації. У межах такого підходу передавання прихованих бітів може здійснюватися шляхом варіювання ком, крапок з комами, тире або інших синтаксичних елементів без зміни загального змісту тексту. Водночас подібні методи мають обмежену область застосування, оскільки надмірні або нетипові пунктуаційні конструкції можуть привертати увагу читача.

Семантичні методи, на відміну від синтаксичних, використовують властивості лексичного складу мови. У таких алгоритмах прихована інформація кодується шляхом заміни слів або словосполучень на синонімічні відповідники. При цьому кожен варіант слова може відповідати певному значенню біта або групі бітів. Основною перевагою цього підходу є вищий рівень природності тексту, оскільки зміни менш помітні для людини.

Разом із тим семантичні методи потребують значних мовних ресурсів і ретельного підбору синонімів. Необхідно враховувати контекст, стилістичні особливості та граматичну узгодженість, інакше текст може втратити цілісність або виглядати штучно. Крім того, для різних мов обсяг доступних синонімічних замін може істотно відрізнятися, що впливає на ефективність приховування інформації.

Варто також зазначити, що як синтаксичні, так і семантичні методи є вразливими до автоматизованого редагування тексту. Переформатування,

автоматична корекція орфографії або пунктуації, а також повторне збереження документа в іншому форматі можуть призвести до втрати прихованих даних. Це обмежує практичне застосування таких методів у середовищах, де текст зазнає частих змін.

Попри зазначені недоліки, лінгвістична стеганографія залишається перспективним напрямом досліджень. Її методи можуть бути ефективними в умовах, коли використання інших типів контейнерів є неможливим або недоцільним. Подальший розвиток цього напряму пов’язаний із удосконаленням алгоритмів автоматичного аналізу мови та створенням більш стійких схем кодування прихованої інформації.

2.3 Методи довільного інтервалу

Існує щонайменше дві причини, через які маніпулювання вільним простором у певних випадках демонструє досить задовільні результати.

По-перше, варіювання кількості пробілів у кінці рядка практично не впливає на зміст речення чи фрази.

По-друге, пересічний читач зазвичай не звертає уваги на незначні зміни в розташуванні вільного простору на текстовій сторінці.

Загально відомими є методи, у яких для приховування інформації використовується вільний простір тексту. Ці методи грають інтервалами між:

– відстанями між словами в тексті, вирівняному за шириною.

– відстанями між словами в тексті, не вирівняному за шириною.

– пробілами в кінці рядків;

– реченнями;

– рядками в абзаці;

– абзацами.

Для визначеності, у подальшому, передбачаємо обробку ASCII-файлів.

Це файли, що:

– містять лише символи кодової таблиці ASCII;

– складаються з рядків, розділених двома символами:

а) кінець рядку (LF);

б) повернення каретки (CR).

Файл, у якому ховаються дані, називається контейнером.

Файл, який підлягає приховуванню, може мати довільний формат, однак з метою спрощення реалізації, розглянемо випадок, коли він є текстовим. За цього визначимо, що такий файл містить лише один рядок, у якому міститься лише одне речення.

На наступному етапі необхідно зчитати файл з інформацією, що підлягає приховуванню, у байтовому режимі. Читання здійснюється послідовно, байт за байтом, кожен з яких може набувати значень у діапазоні від 0 до 255. Під час обробки кожного байта його десяткове значення переводиться у двійкову систему числення, після чого отримані двійкові символи накопичуються в окремому рядку, який у підсумку міститиме двійкове подання всього файлу.

Наприклад. Нехай маємо текстову фразу "АБВ". Під час зчитування першого байта, що відповідає символу "А", отримуємо значення 0x80. Після перетворення в двійкову систему числення воно набуває вигляду 10000000. Оскільки кількість розрядів повинна дорівнювати восьми, зліва дописуються відсутні нулі, які не впливають на значення числа, у результаті чого маємо 10000000. Далі зчитується другий байт – символ "Б" зі значенням 0x81, який у двійковому вигляді записується як 10000001. Аналогічно обробляється третій байт із символом "В", що має значення 0x82 і двійкове подання 10000010. У результаті формується рядок "100000001000000110000010", який є двійковим поданням файлу з вмістом "АБВ".

Застосовуючи описаний підхід, можна отримати двійкове подання будь-якого файлу, що зберігається на диску. Саме таким чином і задається інформація, яку необхідно приховати. Після формування даних для вбудовування можна переходити безпосередньо до розгляду методів

приховування інформації.

2.3.1 Метод зміни інтервалу між реченнями

Метод варіювання інтервалів між реченнями дає змогу вбудовувати повідомлення у двійковому форматі шляхом використання одного або двох пробілів після символів завершення речень. Як такі символи можуть виступати, наприклад, крапки у звичайному тексті або знаки закінчення команд (наприклад, двокрапка чи кома з крапкою) у текстах програм достатньо великої кількості мов програмування. У цьому випадку один пробіл може відповідати біту "1", а два пробіли – біту "0".

Поряд із простотою реалізації цей метод має низку суттєвих недоліків. Насамперед він характеризується низькою ефективністю, оскільки для приховування невеликої кількості інформації потрібен текст значного обсягу. Так, один прихований біт на одне речення відповідає швидкості передавання приблизно один біт на 140 байтів тексту за умови, що середнє речення займає два рядки по 70 символів.

Крім того, можливість застосування цього методу значною мірою залежить від структури текстового контейнера. Деякі тексти, наприклад вільні вірші або верлібри, не мають чітко визначених і стабільних символів завершення рядків або речень.

Ще одним недоліком є використання текстових редакторів, які автоматично додають один або два пробіли після крапки наприкінці речення (функція автозавершення). Нарешті, непослідовне або нестандартне використання пробілів може бути достатньо помітним для уважного читача.

Для реалізації цього методу потрібен файл-контейнер, який містить велику кількість рядків, у кінці яких відсутній пробіл. Це пояснюється тим, що для приховування одного біта інформації (0 або 1) використовується один

рядок контейнера. Оскільки вихідне повідомлення довжиною 10 символів містить 10·8=120 біт інформації, контейнер повинен містити не менше ніж 10×8 рядків.

Після зчитування файлу з приховуваною інформацією та формування двійкової послідовності починається поетапне читання файлу-контейнера по рядках.

Приховування інформації.

Для вбудовування повідомлення послідовно рухаємося по двійковому рядку зліва направо, одночасно зчитуючи по одному рядку контейнера. Якщо поточний біт дорівнює 1, у кінець відповідного рядка додається пробіл, якщо ж біт дорівнює 0 – рядок залишається без змін. Отриманий рядок записується до третього файлу, відкритого для запису.

Процес приховування продовжується доти, доки не буде використано всю двійкову послідовність нулів і одиниць.

Витягування інформації.

Для витягування прихованого повідомлення необхідно послідовно зчитувати рядки контейнера, аналізуючи наявність або відсутність пробілу в кінці кожного з них. Якщо рядок завершується пробілом, це інтерпретується як біт "1", у разі відсутності пробілу фіксується біт "0". У такий спосіб формується двійкова послідовність, яка є результатом зчитування всього контейнера або його визначеної частини.

Отриманий двійковий рядок далі розбивається на групи по вісім бітів. Кожна така група інтерпретується як окремий байт. Для кожного байта виконується перетворення з двійкової системи числення у десяткову, після чого відповідне числове значення трактується як код символу. Сукупність отриманих символів у тій самій послідовності відтворює початковий файл з прихованою інформацією.

Слід зазначити, що для коректного витягування повідомлення необхідно точно знати довжину прихованої двійкової послідовності або мати додаткові ознаки завершення повідомлення. У протилежному випадку процес

відновлення може бути ускладнений через наявність зайвих рядків у контейнері, які не містять закодованих бітів.

Основною перевагою такого методу є простота його реалізації та відсутність складних обчислень. Для вбудовування і витягування інформації достатньо виконувати елементарні операції з текстовими рядками. Разом із тим такий підхід має низку обмежень, пов’язаних із низькою пропускною здатністю та вразливістю до автоматичного форматування тексту.

Зокрема, будь-яке редагування файлу-контейнера, яке призводить до видалення пробілів у кінці рядків, автоматичного вирівнювання тексту або перетворення формату файлу, може повністю зруйнувати приховане повідомлення. Це істотно обмежує можливість практичного застосування методу в умовах, де контейнер підлягає повторному збереженню або обробці стандартними текстовими редакторами.

Таким чином, розглянутий спосіб приховування інформації доцільно використовувати лише в контрольованих умовах, де структура текстового контейнера залишається незмінною протягом усього циклу передавання та зберігання даних.

2.3.2 Метод зміни інтервалу між словами

Цей спосіб приховування інформації також використовує текстовий контейнер, однак принцип кодування бітів у ньому відрізняється від попереднього. У цьому випадку для передавання одного біта інформації застосовується не кінець рядка, а кількість пробілів між окремими словами в межах одного рядка. Таким чином, інформація розподіляється безпосередньо всередині тексту, що зменшує ймовірність її втрати під час незначного редагування.

Для реалізації методу необхідно, щоб контейнер містив достатню

кількість рядків з принаймні двома словами. Кожен такий рядок може бути використаний для приховування одного біта інформації. Якщо між словами використовується один пробіл, це інтерпретується як біт "0", тоді як наявність двох пробілів відповідає біту "1". При цьому зовнішній вигляд тексту для більшості читачів залишається практично незмінним.

Приховування інформації.

Після формування двійкового подання приховуваного повідомлення здійснюється послідовне зчитування рядків контейнера. Для кожного біта вибирається черговий рядок з необхідною кількістю слів. Залежно від значення біта між словами встановлюється один або два пробіли. Модифікований рядок записується у вихідний файл, який і є стеганограмою.

Витягування інформації.

Для відновлення прихованого повідомлення контейнер зчитується пострічково, після чого аналізується кількість пробілів між словами. Визначене значення (один або два пробіли) перетворюється у відповідний біт. Отримана двійкова послідовність далі обробляється стандартним способом: вона розбивається на байти, кожен з яких інтерпретується як код символу, що дозволяє відновити початковий файл.

Основною перевагою цього методу є дещо вища стійкість до часткових змін контейнера в порівнянні з першим способом. Однак він також залишається вразливим до автоматичного форматування тексту, зокрема до вирівнювання по ширині або заміни кількох пробілів на один.

2.3.3 Метод зміни позицій символів у рядку

Третій підхід до приховування інформації в текстових контейнерах базується на використанні позицій символів у рядку. У цьому методі кодування бітів здійснюється шляхом зміни положення окремих символів або незначної

модифікації структури рядка без зміни його змісту з точки зору читача.

Для реалізації такого способу необхідно, щоб контейнер мав чітко визначену структуру та допускав незначні зсуви символів, які залишаються непомітними при звичайному перегляді тексту. Наприклад, можна використовувати варіації розміщення табуляцій, різну кількість пробілів у фіксованих позиціях або інші елементи форматування, які не впливають на семантику повідомлення.

Приховування інформації.

Після отримання двійкової послідовності приховуваного повідомлення зчитування контейнера здійснюється по рядках. Для кожного біта вибирається відповідний рядок, у якому залежно від значення біта виконується визначена модифікація позиції символів. Наприклад, незначний зсув символа вправо може відповідати біту "1", тоді як відсутність зсуву – біту "0". Змінений рядок записується до вихідного файлу.

Витягування інформації.

Під час відновлення прихованих даних контейнер аналізується з урахуванням позицій символів у кожному рядку. Фіксуються наявність або відсутність відповідних зсувів, які інтерпретуються як біти двійкової послідовності. Далі процедура відновлення байтів і символів виконується аналогічно попереднім методам.

Перевагою цього підходу є можливість гнучкішого використання форматувальних особливостей тексту. Проте він також має суттєві недоліки, зокрема високу чутливість до переформатування документа, зміни шрифту або конвертації в інші формати, що може призвести до втрати прихованої інформації.

2.3.4 Метод заміни символів

Для застосування цього способу потрібен файл-контейнер, у якому міститься значна кількість кириличних літер, графічне зображення яких збігається або є дуже подібним до відповідних латинських символів. До таких літер належать, зокрема, А, К, Е, Р, В, М, Н, е, р, о, с тощо. Допускається використання як малих, так і великих літер. Кількість подібних символів у контейнері має щонайменше у вісім разів перевищувати кількість літер у повідомленні, яке необхідно приховати. Причина цього обмеження аналогічна тій, що зазначалася для попередніх методів.

Приховування інформації.

Після зчитування файлу з інформацією, яка підлягає маскуванню, та формування її двійкового представлення починається посимвольне читання файлу-контейнера. Для вбудовування всього повідомлення здійснюється послідовний рух зліва направо по двійковому рядку. Кожен черговий символ контейнера аналізується. Якщо кирилична літера не має латинського графічного аналога, вона залишається без змін і записується до третього файлу. Якщо ж символ має відповідник в латинському алфавіті, здійснюється перевірка поточного біта двійкової послідовності. У разі, якщо біт дорівнює 1, кирилична літера замінюється на відповідну латинську. Якщо значення біта дорівнює 0, символ залишається без змін. Отриманий символ записується до вихідного файлу. Така процедура повторюється до повного використання всієї послідовності нулів і одиниць.

Витягування інформації.

На вході маємо файл, що складається з символів. Його зчитування виконується послідовно, символ за символом. Кожен елемент аналізується. Якщо кирилична літера не має латинського аналога, вона ігнорується, і здійснюється перехід до наступного символу. Якщо ж відповідник існує,

необхідно визначити, до якого алфавіту належить поточний символ. Для цього використовується код символу в таблиці ASCII.

Як зазначалося раніше, кожен символ має числовий код у діапазоні від 0 до 255. Наприклад, кирилична літера «В» має код 130, тоді як латинська літера «B» відповідає коду 42. Слід враховувати, що великі й малі літери одного й того самого алфавіту кодуються по-різному.

Для визначення числового коду конкретного символу застосовується функція ord(X), де X – заданий символ.

Функція ord(X) повертає числове значення символу в межах від 0 до 255.

Натомість функція chr(d) виконує зворотну операцію та повертає символ, що відповідає заданому числу d.

Повертаючись до процесу витягування, зазначимо, що якщо значення, отримане за допомогою ord(X), потрапляє до інтервалу кодів латинських літер (наприклад, від 40 до 91), то відповідний біт двійкової послідовності приймає значення 1. В іншому випадку він вважається рівним 0. Після повного зчитування вхідного файлу формується двійковий рядок. Далі ця послідовність розбивається на групи по вісім бітів (байти) та переводиться з двійкової системи числення в десяткову. Отримані значення інтерпретуються як символи й послідовно записуються до нового вихідного файлу. Саме цей файл і міститиме повідомлення, яке було приховано в початковому файлі-контейнері.

2.3.5 Метод зміни порядку CR/LF

Метод зміни порядку проходження маркерів кінця рядка CR/LF використовує індиферентність гнітючого числа засобів відображення текстової інформації до порядку проходження символів перекладу рядка (CR) і повернення каретки (LF), що обмежують рядок тексту. Традиційний порядок проходження CR/LF відповідає 0, а інвертований LF/CR означає 1.

2.3.6 Метод додавання Unicode-символів

Використання Unicode-символів дозволяє істотно розширити кількість символів, які замінюють пробіл на інший символ, схожий на пробіл (табл. 2.1).

Але не усі пробіли є видимі: деякі не відображаються на екрані (мають нульову ширину). Через це, не усі можна використовувати в якості багаторозрядної заміни стандартного (розділового між символами) пробілу.

У тих алгоритмах, де Unicode-пробіл використовується як розділовий символ, застосовують видимі пробіли. Їх виділено жирним шрифтом у таблиці 2.1.

У тих алгоритмах, де Unicode-пробіл використовується для вставки прихованих (невидимих) символів, застосовують невидимі пробіли. Їх наведено звичайним шрифтом у таблиці 2.1.

Таблиця 2.1 – Пробіли

Назва символу	Код	Видимий	.txt	.docx	.pdf	.eml
пробіл	OxOO2O	+	+	+	+	+
нерозривний пробіл	OxOOAO	+	-	+	+	-
пробіл з кельтського набору символів	Ox168O	-	+	+	+	+
En довгий пробіл	Ox2OOO	-	+	+	+	+
Em довгий пробіл	Ox2OO1	-	+	+	+	+
En пробіл	Ox2OO2	-	+	+	-	+
Em пробіл	Ox2OO3	-	+	+	-	+
3x-пробіл	Ox2OO4	+	+	+	+	+
4x-пробіл	Ox2OO5	-	+	-	-	-
6x-пробіл	Ox2OO6	-	+	+	+	+
фігурний пробіл	Ox2OO7	-	+	+	+	+

Кінець таблиці 3.3

Назва символу	Код	Видимий	.txt	.docx	.pdf	.eml
синтаксичний пробіл	Ox2OO8	+	+	+	+	+
тонкий пробіл	Ox2OO9	+	+	+	+	+
надтонкий пробіл	Ox2OOA	-	+	+	+	+
тонкий нерозривний пробіл	Ox2O2F	+	+	+	+	+
середній математичний пробіл	Ox2O5F	+	+	+	+	+
ідеографічний пробіл	Ox3OOO	-	+	+	-	+

Критерій видимості або невидимості виробляється за підсумками порівняння ширини пробільних символів із стандартним пробілом з кодом 0x0020. Ширина останнього, приблизно, дорівнює 1/4 em. За наявності відмінностей, які виникають через застосування нестандартного пробілу, символ розглядається як такий, що має незадовільну видимість, позначаючи його символом "-" у таблиці 2.1.

Випадок, коли різниця непомітна, позначається символом "+", що відповідає, приблизно, ширині від 1/5 до 1/3 em). Натомість, в іншому випадку, "-" вказує, що різниця може бути помітною для людського ока (наприклад, 1/6 або 1/2 em).

Лише:

– пробіл потрійної ширини (Ox2OO4);

– розділовий пробіл (Ox2OO8);

– вузький пробіл (Ox2OO9);

– вузький нерозривний пробіл (Ox2O2F);

– середній математичний пробіл (Ox2O5F), не помітні людиною і можуть вільно застосовуватись у більшості програм й форматів файлів.

3 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

3.1 Розробка алгоритму

Відповідно до технічного завдання на дипломне проектування, за розробки алгоритму задамося метою не збільшування обсягу тексту. Тобто, між словами мусить використовуватися лише один розділовий символ.

Відповідно до технічного завдання на дипломне проектування, за розробки алгоритму орієнтуватимемося лише на сучасні текстові редактори із складу наступних пакетів:

– Microsoft Office;

– Open Office;

– WPS Office;

– Libre Office, а також на файли формату pdf.

Відповідно до технічного завдання на дипломне проектування, застосуємо метод заміни символу пробілу як ознаку інтервалу між словами.

Цей спосіб приховування інформації використовує текстовий контейнер, однак принцип кодування бітів у ньому відрізняється від розглянутих. У цьому випадку для передавання одного біта інформації застосовується не кінець рядка, не кількість пробілів між окремими словами в межах одного рядка, а лише використання символів пробілів з різними кодами. Таким чином, інформація розподіляється безпосередньо всередині тексту, що зменшує ймовірність її втрати під час незначного редагування.

Для реалізації методу необхідно, щоб контейнер містив достатню кількість рядків з принаймні двома словами. Кожен такий рядок може бути використано для приховування одного біта інформації. Якщо між словами використовується пробіл 0x20, це інтерпретується як біт "0", тоді як наявність пробілу з іншим кодом відповідає біту "1". При цьому зовнішній вигляд тексту для більшості читачів залишається практично незмінним.

Приховування інформації.

Після формування двійкового подання приховуваного повідомлення здійснюється послідовне зчитування слів контейнера. Для кожного біта береться поточний абзац з довільною кількістю пробілів. Залежно від значення біта інформаційного повідомлення, між словами встановлюється той чи інший символ пробілу. Модифікований абзац записується у вихідний файл, який і є стеганограмою.

Витягування інформації.

Для відновлення прихованого повідомлення контейнер зчитується пробільно, після чого визначається код символу пробілу між словами. Визначене значення (той чи інший пробіли) перетворюється на відповідний розряд. Отримана двійкова послідовність далі обробляється стандартним способом: вона розбивається на байти, кожен з яких інтерпретується як код символу, що дозволяє відновити початкове інформаційне повідомлення.

Основною перевагою цього методу є дещо вища стійкість до часткових змін контейнера в порівнянні з відомим способом. Він не залишається вразливим до автоматичного форматування тексту, зокрема до вирівнювання по ширині або заміни кількох пробілів на один.

3.2 Розробка програми

Розробку програми здійснюватимемо мовою Delphi, здійснюючи обробку doc-файлів засобами Automation шляхом застосування стандартних компонентів для роботи с OLE/COM, а також методи й властивості об’єктної моделі Microsoft Word (наприклад, Documents, Paragraphs, Range тощо (рис. 3.1).

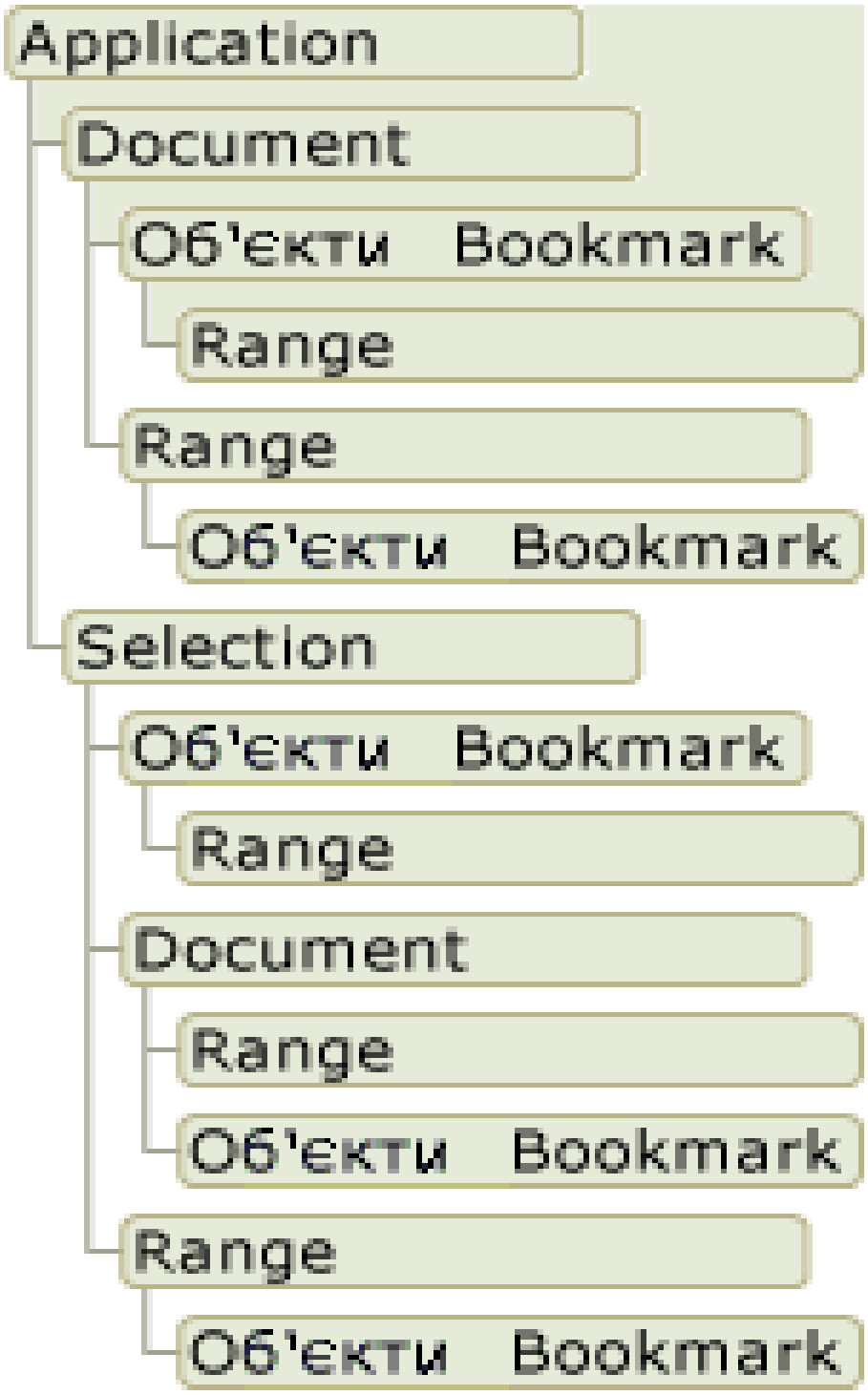


Рисунок 3.1 – Об’єктна модель Microsoft Word

Об'єкт Application – це основний інтерфейс у компонентному класі COM, що визискується для взаємодії програмного коду з відповідним об'єктом COM стороннього додатка.

Як основний інтерфейс, об'єкт Application використовується тільки в тих

випадках, коли метод, що передбачається використовувати, є одноіменним з подією об'єкта COM.

Об'єкт Document є членом одноіменної колекції Documents. Колекція Documents містить всі об'єкти Document, що у даний час відкриті в Microsoft Word.

Об'єкт Selection являє собою виділену чи обрану область у документі. Об'єкт Selection являє собою крапку вставки, якщо в документі нічого не обрано.

У кожному документі з колекції Documents може бути тільки один об'єкт Selection.

Відповідно, в усьому додатку в один момент часу тільки один об'єкт Selection може бути активним.

Властивість Selection використовується, щоб повернути об'єкт Selection. Microsoft Word повертає виділення з активної області поточного вікна документа.

Властивість Text об'єкта Selection використовується для доступу до тексту у поточному виділеному фрагменті.

Об'єкт Selection має різні методи і властивості.

Об'єкт Bookmark є членом одноіменної колекції Bookmarks. Колекція Bookmarks містить усі закладинки, перелічені в діалоговому вікні "Закладка" пункту меню "Вставка" додатка Microsoft Word.

Доступ до закладок здійснюється через Bookmarks(index), де index – це ім'я чи закладинки номер індексу, що вказує на один об'єкт Bookmark.

Номер індексу визначає позицію закладинки в об'єкті Selection чи Range об'єкта Document.

Для об'єкта Bookmark, номер індексу представляє позицію закладинки в переліку закладинок діалогового вікна "Закладки" додатка Microsoft Word.

Кожен об'єкт Range визначається:

– початковою;

– кінцевою, позиціями символів.

Як і об'єкти Bookmark, об'єкти Range використовуються для доступу до визначених частин документа. Однак, на відміну від об'єкта Bookmark, об'єкт Range існує тільки під час виконання програмного коду.

Об'єкти Range не залежать від виділеного фрагмента тексту. Вони використовуються для:

– визначення діапазону;

– керування діапазоном тексту.

Але для спрощення процедури відлагоджування алгоритмів, використовуватимемо Turbo Pascal.

Для створення програми, необхідно розробити її інтерфейс й, принаймні, два алгоритми:

– приховування;

– видобування даних з текстового файлу.

3.2.1 Реалізація відлагоджувального алгоритму приховування інформаційного повідомлення до текстового файлу

```
program p;
uses crt,math;
var f1,f2,f3,f4:text;
   i,j,k1,n:integer;
   s1,s2,s3,ss:string;
   ch,ch3:char;
begin
 clrscr;
 assign(f1,'y:\1\11.txt');
 reset(f1);
 assign(f2,'y:\1\21.txt');
```

```
reset(f2);
assign(f3,'y:\1\31.txt');
rewrite(f3);
n:=1;
repeat
  readln(f1,s1);
  {if not eof(f1) then}
    begin
    {if s1[length(s1)]<>' ' then s1:=s1+' ';}
    k1:=Pos(' ',s1)-1;
    while k1>0 do
     begin
      ch3:=' ';
      if n=1 then
       begin
        read(f2,ch);
        if not eof(f2) then
         begin
          write(ch);
          case Ch of
            #13,#10:begin
                      read(f2,ch);
                      read(f2,ch);
                      writeln;
                     end;
          end;
          n:=ord(ch);
         end;
       end;
      ch3:=' ';
```

```
      if (n mod 2)=1 then ch3:=#9;
      n:=n div 2;
      ss:=copy(s1,1,k1)+ch3;
      write(f3,ss);
      s1:=copy(s1,k1+2,5);
      k1:=Pos(' ',s1)-1;
     end;
    writeln(f3,s1);
    {writeln;}
    {repeat
      readln(f1,s);
      readln(f2,ss);
      writeln(f3,s);
    until eof(f1) or eof(f2);}
   end;
  {stop;}

 until eof(f1) or eof(f2);

 close(f1);
 close(f2);
 close(f3);
end.
```

3.2.2 Реалізація відлагоджувального алгоритму видобування інформаційного повідомлення з текстового файлу

```
program p;
uses crt,math;
var f1,f2,f3,f4:text;
    i,j,k1,n:integer;
    s1,s2,s3,ss:string;
    ch,ch3:char;
begin
 clrscr;
 assign(f3,'y:\1\31.txt');
 reset(f3);
 assign(f4,'y:\1\41.txt');
 rewrite(f4);
 clrscr;
 n:=0;
 k1:=0;
 repeat

   readln(f3,s3);
   {writeln(s3);}


   for i:=1 to length(s3) do
    begin
     if s3[i]=' ' then
      begin
```

```
      {k:=k or 1 lsh n;}
      n:=n+1;
      {write('0');}
     end;
    if s3[i]=#9 then
     begin
      k1:=k1 or (n shr 1);
      n:=n-1;
      {write('1');}
     end;
    if n=512 then
     begin
      write(f4,chr(k1));
      {break;}
      write(chr(k1));
      n:=0;
      k1:=0;
     end;
   end;
 until eof(f3);
 if n<>0 then write(f4,chr(k1));
 close(f3);
 close(f4);
end.
```

3.3 Програми-аналоги

Порівняння розроблених алгоритмів здійснюватимемо на відомих

програмах стеганографічного захисту тексту.

Через зростаючу різноманітність методів, носіїв даних та застосувань, а також появу численних аналогів, у яких втілено один й той самий метод і які використовують приблизно однакові техніки вставки або підстановки символів, цікаво вибрати з наявних стеганографічних сервісів.

Вимоги до наявних стеганографічних сервісів:

– безкоштовність;

– відсутність необхідності реєстрації для доступу;

– без обмежень на багаторазовість використання.

На жаль, наявних стеганографічних сервісів, які б задовольняли висунутим вимогам, виявилось обмаль.

3.3.1 SNOW

Steganographic Nature of Blankspace (SNOW) є одним із найдавніших засобів приховування інформаційних повідомлень у порожніх місцях рядка.

Численні варіанти програми SNOW під різні платформи працюють з 2013 р. під відкритою ліцензію Apache 2.0. Вона має версії під:

– MS DOS;

– Windows;

– Linux, а також реалізацію під Java (останнє оновлення – у 2016 році).

Програма SNOW здійснює процес вбудовування інформаційного повідомлення у символи tab і пробіли та додає їх до файлу-контейнера (тексту обгортки), починаючи з символу tab з урахуванням заздалегідь визначеної довжини рядка.

Дозволяє здійснення:

– стиснення;

– шифрування, які можна увімкнути до початку процесу кодування.

Програма доступна за адресою: https://darkside.com.au/snow/

3.3.2 StegZero

StegZero – Web-сервіс, який ховає повідомлення, використовуючи символи нульової ширини.

Стеганографічний агоритм StegZero утаємничує інформаційні повідомлення за допомоги байтів кодової таблиці UTF-8, забезпечуючи повну підтримку символів Unicode.

Стеганографічний агоритм StegZero маскує тільки біти інформаційного повідомлення, використовуючи генератор псевдовипадкових чисел з попереднім ініціалізацією на основі:

– шифрування (для комерційного використання);

– одноразового числа (для некомерційного використання), щоб спростити виявлення шаблонів (з демонстраційною метою).

В основі стеганографічного агоритму StegZero лежить механізм перетворення трирозрядних груп на різноманітні набори з восьми невидимих символів (символів нульової ширини), а не просто одну пару "0"/"1". Сховані символи рівномірно розподіляються текстом, а не накопичуються наприкінці файлу.

Заявлено, що реалізація стеганографічного агоритму StegZero містить у собі резервний декодер для більш старих стеганографічних агоритмів, у яких використовувалася проста двохсимвольная схема ("0"/"1").

Web-сервіс доступний за адресою: https://stegzero.com/

3.4 Дослідження зростання обсягів

На рисунку 3.2 наведено розмір файлу L (у байтах) в залежності від кількості символів інформаційного повідомлення N.

Оскільки метод заміни символу пробілу не передбачає внесення додаткової інформації до файлу, то обсяг вихідного файлу не залежить від кількості символів інформаційного повідомлення.

Оскільки алгоритм роботи програми SNOW передбачає додавання необхідної кількості пробілів в кінець рядка, то обсяг вихідного файлу залежить від кількості символів інформаційного повідомлення: чим довше інформаційне повідомлення – тим більше файл.

Оскільки програма власної розробки й програма SNOW тестувались на однакових обсягах файлу-контейнера й інформаційного повідомлення, то вони починаються з однієї крапки на графіку.

На жаль, Web-сервіс StegZero було знайдено і, відповідно, протестовано, пізніше (коли дослідження двох попередніх програм вже відбулися). Під час дослідження роботи некомерційної реалізації алгоритму StegZero з’ясувалось, що:

– роботу з файлами не передбачено;

– можна працювати лише з короткою фразою-контейнером;

– можна працювати лише з короткою фразою інформаційного повідомлення.

Тому алгоритм StegZero досліджувався на меншому обсязі контейнера. Через це, відповідна крива проходить значно нижче двох попередніх.

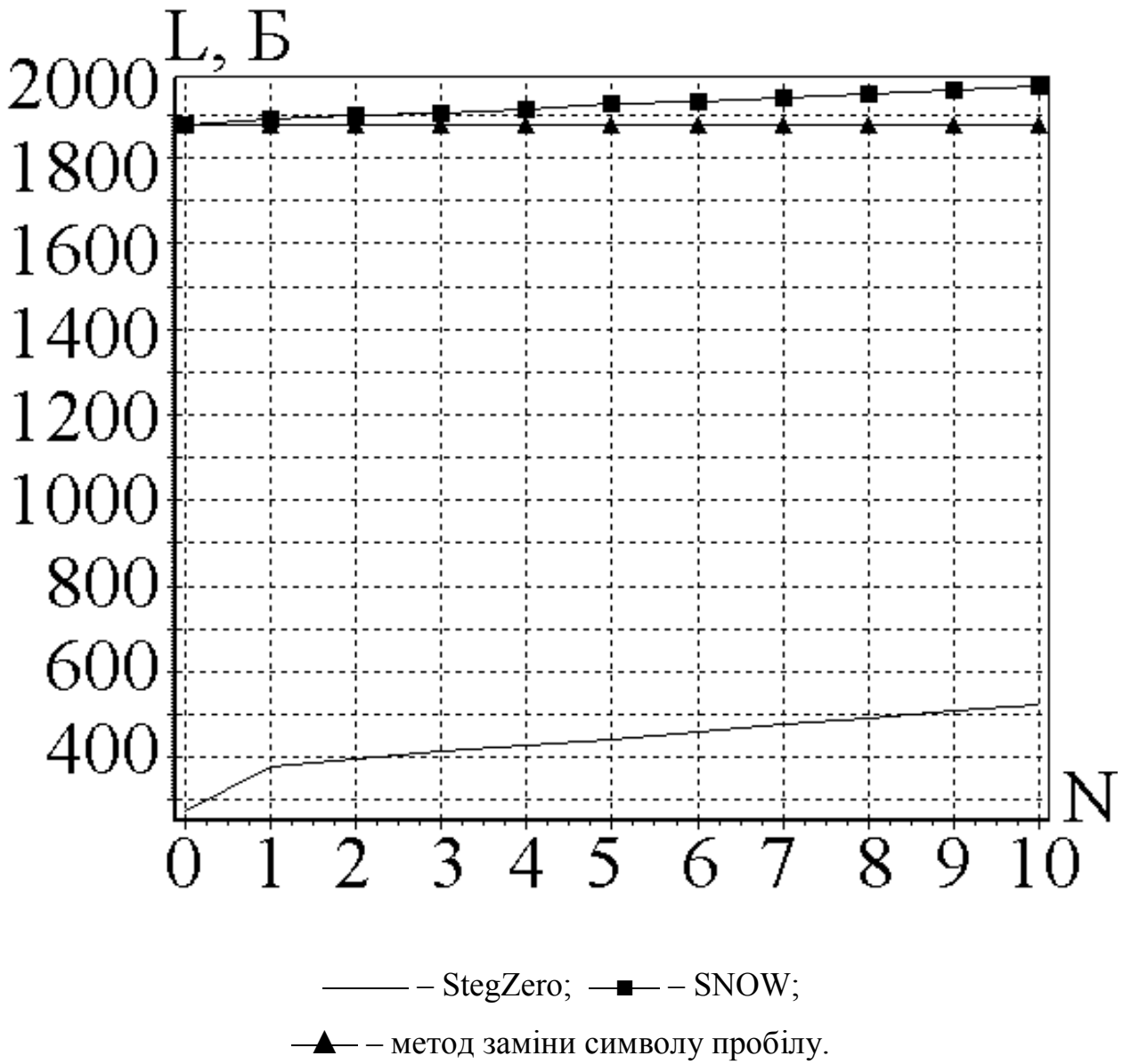


Рисунок 3.2 – Зміна розміру файла L в залежності від збільшення кількості символів інформаційного повідомлення N

На рисунку 3.3 наведено зміни обсягу контейнера (кількості доданих символів M) в залежності від кількості символів інформаційного повідомлення N.

Оскільки метод заміни символу пробілу не передбачає внесення додаткової інформації до файлу, то обсяг вихідного файлу не залежить від кількості символів інформаційного повідомлення.

Оскільки алгоритм роботи програми SNOW передбачає додавання необхідної кількості пробілів в кінець рядка, то обсяг вихідного файлу збільшується, приблизно, на 8 символів на кожний символ інформаційного повідомлення. Точніше:

– кожні перші два символи інформаційного повідомлення збільшують обсяг файла на 9 символів;

– кожний третій символ інформаційного повідомлення збільшує обсяг файла на 5 символів.

Тому алгоритм StegZero досліджувався на меншому обсязі контейнера. Через це, відповідна крива проходить значно нижче двох попередніх.

Оскільки алгоритм StegZero передбачає додавання деякої кількості пробілів у певне місце контейнера, то обсяг вихідного файлу збільшується, приблизно, на 8 символів на кожний символ інформаційного повідомлення. Точніше:

– кожний перший символ інформаційного повідомлення збільшує обсяг файла на 9 символів;

– кожний другий символ інформаційного повідомлення збільшує обсяг файла на 8 символів.

За цього, додавання першого символу супроводжується додаванням 44 символів службової інформації.

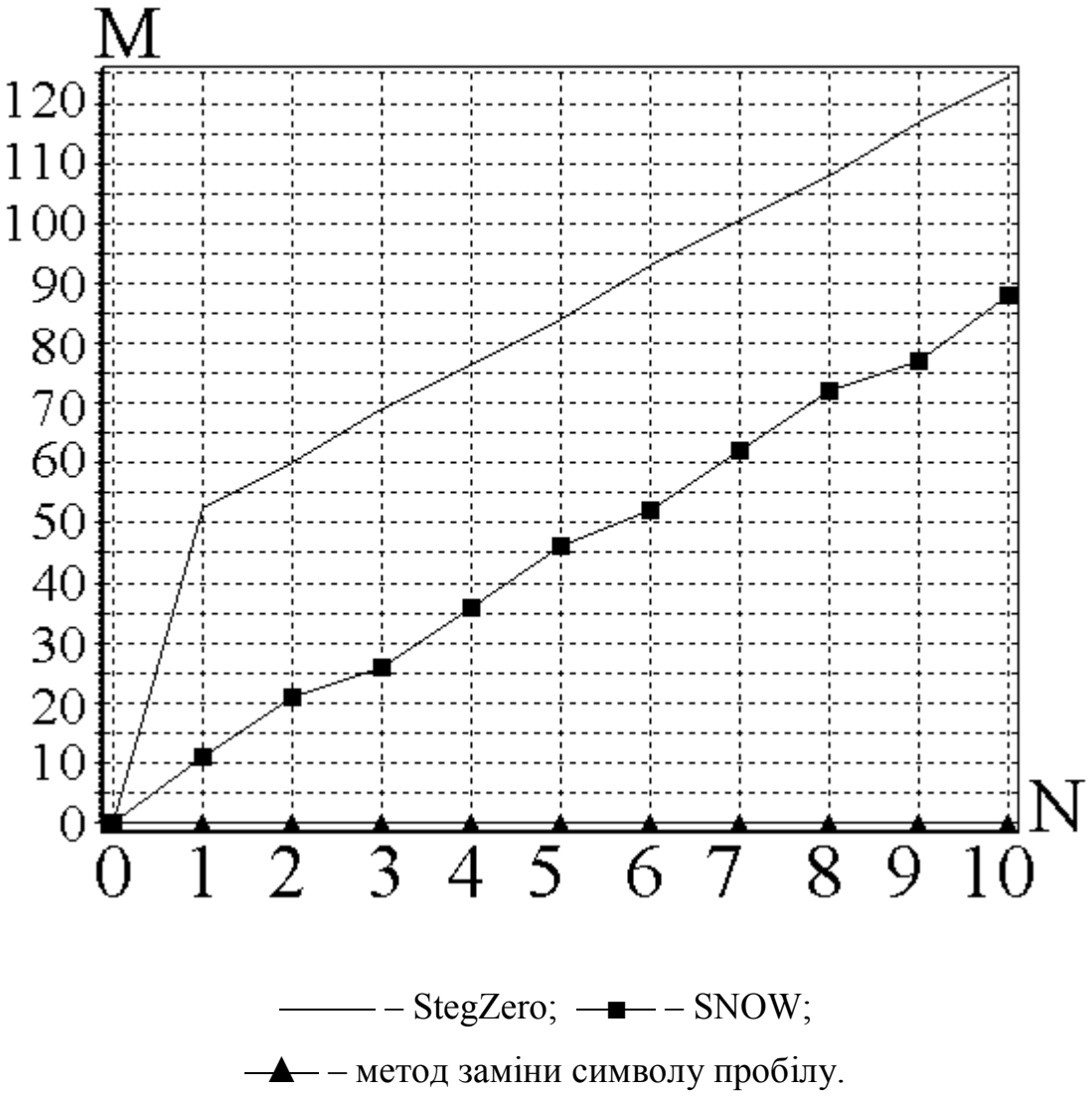


Рисунок 3.3 – Зростання обсягу контейнера (кількості символів M) в залежності від збільшення кількості символів інформаційного повідомлення N

3.5 Дослідження на стеганостійкість

На своєму сайті, сервіс StegZero обіцяє дуже гарний результат (рис. 3.4): зовнішній вигляд рядка-контейнера (поле "Visible Text") нічим не відрізняється від результату (поле "Result") після вбудовування до останнього однієї прихованої літери (поле "Hidden Message").

Але, вставивши результат (через буфер обміну) до текстового редактора

Microsoft Word, можна побачити дві великі різниці (рис. 3.5) між звичайним реченням й модифікованим реченням. Такий зовнішній вигляд свідчить про:

– задовільну стеганостійкість у WEB-застосунках;

– незадовільну стеганостійкість у doc й pdf-файлах.

Вставивши результат модифікації рядка за допомоги алгоритму SNOW (через буфер обміну – рис. 3.6) до текстового редактора Microsoft Word, можна зробити висновок про забезпечення стеганостійкості у doc й pdf-файлах. Але особливість реалізації алгоритму (один символ на один абзац) вимагає великих обсягів тексту-контейнера.

Вставивши результат модифікації рядка за допомоги розробленого в ході дипломування алгоритму (через буфер обміну – рис. 3.7) до текстового редактора Microsoft Word, теж можна зробити висновок про забезпечення стеганостійкості у doc й pdf-файлах. Аби різна відстань між словами у випадку вирівнювання за шириною (рис. 3.7, б)) не так кидалась в очі, можна використовувати вирівнювання абзаців за лівим краєм (рис. 3.7, в)).

Рисунок 3.4 – Web-сервіс StegZero

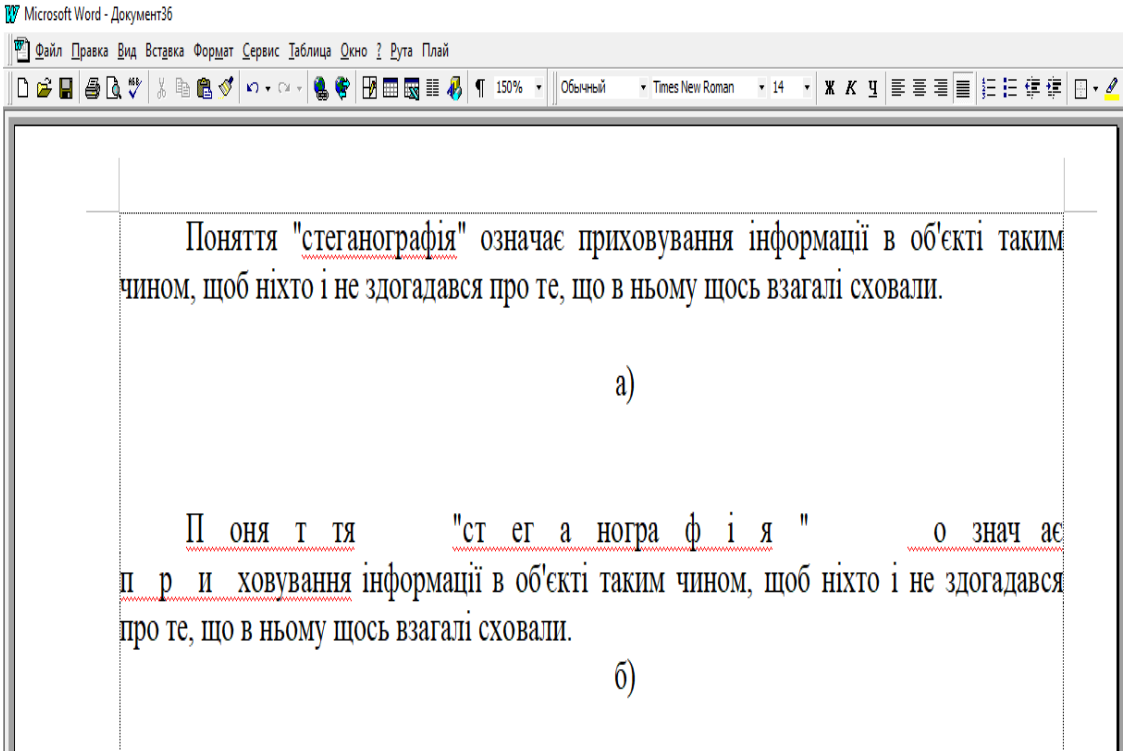


Рисунок 3.5 – Зовнішній вигляд речення у текстовому редакторі Microsoft Word до (а) й після (б) модифікації за допомоги сервісу StegZero

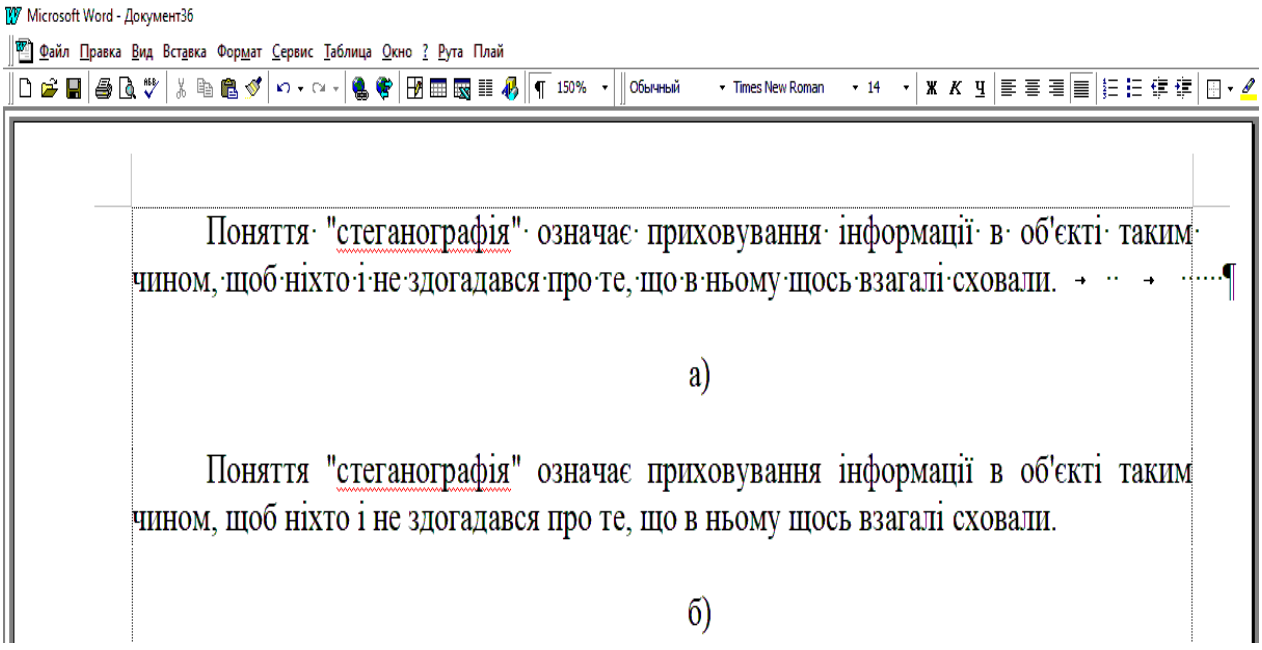


Рисунок 3.6 – Зовнішній вигляд речення у текстовому редакторі Microsoft Word після модифікації за допомоги алгоритму SNOW з увімкненими недрукованими символами (а) й вимкненими (б)

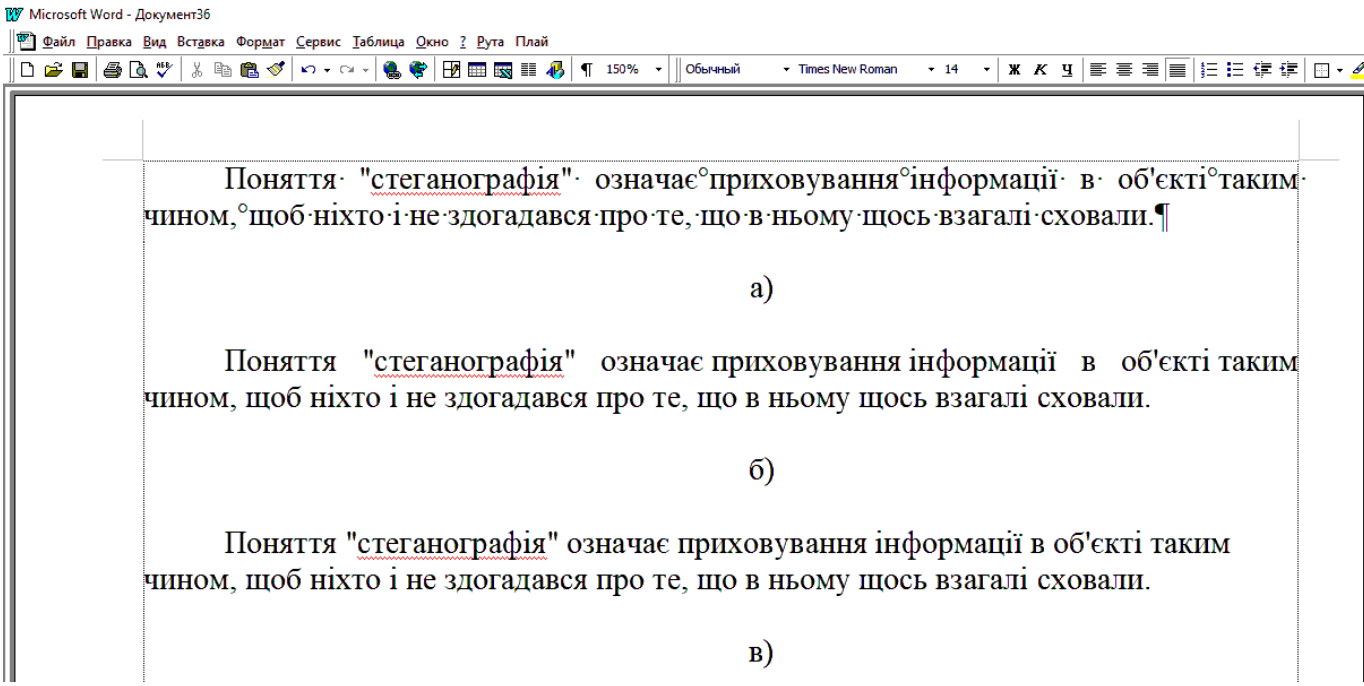


Рисунок 3.7 – Зовнішній вигляд речення у текстовому редакторі Microsoft Word після модифікації за допомоги розробленого алгоритму з увімкненими недрукованими символами (а) й вимкненими недрукованими символами (б, в); з вирівнюванням за шириною (б) й з вирівнюванням за лівим краєм (в)

3.6 Алгоритм шифрування

З метою зменшування обсягу файлу-контейнера, перейдемо до використання третього символу, наприклад, - табуляції.

Задамося питанням: якщо замінювати не один розряд інформаційного повідомлення на символ того чи іншого пробілу, а два розряди одразу, то як часто зустрічатиметься така комбінація в межах символів алфавіту, яким передається інформаційне повідомлення?

Для визначеності (і простоти), обмежимося використанням лише малих літер латинського алфавіту в кодовій таблиці ASCII і для файлу-контейнера, і для інформаційного повідомлення, зміст якого треба утаємничити.

У таблицях 3.1-3.3 наведено імовірності зустрічі дво-, три- й чотирирозрядних комбінацій, відповідно, серед усіх літер алфавіту.

Таблиця 3.1 – Імовірність зустрічі 2 розрядів

Комбінація	Імовірність зустрічі
01b	26%
10b	26%
11b	27%
00b	21%

Таблиця 3.2 – Імовірність зустрічі 2 розрядів

Комбінація	Імовірність зустрічі
001b	12%
010b	9%
011b	17%
100b	12%
101b	14%
110b	18%
111b	9%
000b	7%

Таблиця 3.3 – Імовірність зустрічі 2 розрядів

Комбінація	Імовірність зустрічі
0001b	5%
0010b	3%
0011b	8%
0100b	4%
0101b	5%
0110b	11%

Кінець таблиці 3.3

Комбінація	Імовірність зустрічі
0111b	7%
1000b	5%
1001b	7%
1010b	5%
1011b	9%
1100b	8%
1101b	10%
1110b	7%
1111b	2%
0000b	2%

Аналізуючи таблиці 3.1-3.3, можна побачити, що із збільшенням довжини комбінацій для заміни, зменшується частота їхньої зустрічі. Тобто, якщо імовірність зустрічі дворозрядної комбінації, – приблизно, – як 1 до 4, то для чотирирозрядної комбінації, яка потенційно більш ефективно кодуватиме інформаційне повідомлення, не більше як 1 до 10 (у ліпшому разі)…

Зрозуміло, що у кожному частинному випадку використання тієї чи іншої мови, а також того чи іншого національного алфавіту у:

– файлі-контейнері;

– інформаційному повідомленні, призводитиме до різної ефективності кодування. І тому ні які статистика, оцінка ефективності тощо тут будуть безпредметні. Тобто, у загальному випадку, неможливо вибрати оптимальний спосіб багаторозрядного кодування.

Більш цікавою задачею є підвищення криптостійкості шифру. Для цього треба перейти від кодування із сталою розрядністю протягом усього інформаційного повідомлення (табл. 3.1..3.3) до його кодування із змінною розрядністю для кожної чергової групи розрядів, що підлягають шифруваннєві.

Опишемо найпростіший алгоритм шифрування зі змінною кількістю

розрядів для кодування одного символу інформаційного повідомлення:

а) усе інформаційне повідомлення подається неперервним двійковим кодом;

б) якщо N молодших розрядів символу лівіше від пробілу (опускаючи знаки пунктуації) у файлі-контейнері збігається з групою N послідовних чергових розрядів інформаційного повідомлення, то вона кодується символом табуляції. Перейти до п. 2;

в) якщо молодший розряд символу лівіше від пробілу (опускаючи знаки пунктуації) у файлі-контейнері не збігається із черговим розрядом інформаційного повідомлення, то він кодується парою пробіл або нерозривний пробіл. Перейти до п. 2.

Довжина послідовності розрядів N може визначатись за різними алгоритмами. Наприклад, це – значення 4 молодших розрядів того самого символу лівіше від пробілу. Або операція XOR від 4 молодших розрядів символів навколо поточного пробілу. Тобто, варіантів можна вигадати безліч.

Очевидно, що побудова будь-якої статистики тут теж буде безпредметною…

ВИСНОВКИ

У магістерській роботі розглянуто актуальну проблему захисту текстової інформації шляхом розроблення та дослідження алгоритму шифрування даних у текстових файлах. Проведений аналіз наукових джерел дозволив узагальнити сучасні підходи до симетричного шифрування, принципи криптоаналізу та основні методи стеганографії, що використовуються для приховування інформації в текстових контейнерах.

У процесі дослідження було сформульовано вимоги до нового алгоритму шифрування та розроблено його математичну модель і структурну схему. Реалізовано програмний засіб, який забезпечує шифрування та дешифрування текстових файлів із використанням симетричного ключа, що дозволяє автоматизувати процес захисту інформації та спростити його практичне застосування.

Під час експериментальних досліджень підтверджено працездатність розробленого алгоритму, оцінено його швидкодію та обсяг інформації, який може бути надійно зашифрований і прихований у текстовому файлі. Отримані результати засвідчили, що запропонований підхід забезпечує належний рівень криптостійкості та не поступається відомим аналогам за основними експлуатаційними показниками.

Проведений порівняльний аналіз дозволив визначити переваги розробленого алгоритму, серед яких простота реалізації, універсальність застосування для текстових файлів різних форматів та можливість інтеграції з іншими засобами захисту інформації. Водночас встановлено, що ефективність алгоритму залежить від коректності вибору ключових параметрів і умов обробки текстового контейнера, що визначає напрями подальших досліджень.

Отримані в роботі результати можуть бути використані в навчальному процесі з дисциплін, пов’язаних із криптографією та захистом інформації, а також у практичній діяльності для забезпечення конфіденційності текстових

даних у локальних і мережевих інформаційних системах. Подальший розвиток роботи доцільно спрямувати на вдосконалення алгоритму, підвищення його стійкості до сучасних методів криптоаналізу та розширення сфери застосування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Кузнецов О.О. Стеганографія: навчальний посібник / О.О. Кузнецов, С.П. Євсеєв, О.Г. Король. – Х.: ХНЕУ, 2011. – 232 с.

2. Азаров О. Д., Хорошко В. О., Шелест М. Є., Яремчук Ю. Є. Основи комп'ютерної стеганографії: навчальний посібник. – Вінниця: ВДТУ, 2003. – 143 с.

3. Хорошко В.О. Комп'ютерна стеганографія: навчальний посібник / В.О. Хорошко, Ю.Є. Яремчук, В.В. Карпінець. – Вінниця: ВНТУ, 2017. – 155 с.

4. Хорошко В.О. Основи комп'ютерної стеганографії: навч. посібн. для студентів і аспірантів / В.О. Хорошко, О.Д. Азаров, М.В. Шелест та ін. – Вінниця: ВДТУ, 2003. – 143 с.

5. Конахович Г.Ф. Комп'ютерна стеганографічна обробка й аналіз мультимедійних даних: підручник / Г.Ф. Конахович, Д.О. Прогонов, А.Ю. Пузиренко. – К.: ЦУЛ, 2018. – 555 с.

6. Хорошко В.О. Термінологічний довідник з питань технічного захисту інформації / В.О. Хорошко, І.М. Огаркова, Д.В. Чирков та ін. – К.: ТОВ "ПоліграфКонсалтинг", 2003. – 286 с.

7. Кухарська Н.П. Програмна реалізація алгоритмів приховування інформації методами довільного інтервалу / Н.П. Кухарська // Вісник ЛДУ БЖД. – 2018, № 14. – С. 41-48.

8. Юдін О. Аналіз стеганографічних методів приховування інформаційних потоків у контейнерах різних форматів / О. Юдін, Р. Зюбіна, О. Фролов // Радиоэлектроника и информатика. – 2015, № 3. – С. 13-21.

ДОДАТОК А

Презентація

Національний університет "Запорізька політехніка"
Кафедра інформаційної безпеки та наноелектроніки

Дипломна робота

Алгоритм шифрування інформації в текстовому файлі

Виконав: ст. гр. БКз-814м А. А. Хаметов

Метою дипломної роботи є розробка нового алгоритму утаємничування інформації у текстових файлах.

Для досягнення поставленої мети в ході дипломного проектування, необхідно було розв'язати наступні задачі:

- проведення аналізу ефективності наявних методів шифрування інформації у текстових файлах;
- розробка нового алгоритму утаємничування інформації у текстових файлах.

Методи приховування інформації у текстових файлах:

1. Синтаксичні методи.
2. Семантичні методи.
3. Методи заміни схожих символів.
4. Методи довільного інтервалу:

а) між словами;

б) між рядками;

в) між літерами.

Контейнер - файл, у якому треба сховати деяке інформаційне повідомлення. 3

Сутність методів автоматизованого шифрування:

1. Перетворити інформаційне повідомлення на послідовність логічних нулів й одиниць:

Символ	Код ASCII		
	dec	hex	bin
А	128	0x80	10000000b
Б	129	0x81	10000001b
В	130	0x82	10000010b

"АБВ" = "100000001000000110000010"

2. Вбудувати до файлу-контейнера отриману послідовність за тим чи іншим алгоритмом. 4

Алгоритм заміни схожих символів

A, a, B, C, c, O, o, T, K, M, P, p, …

Символ	Алфавіт	Код	Значення	Застосування
M	кириличний	0x8c	0	файли
M	латинський	0x4d	1	ASCII
M	грецький	0x39c	01	файли
M	кельтський	0x3fa	10	Unicode

Символ	Алфавіт	Код	Значення	Застосування
C	кириличний	0x91	0	файли
C	латинський	0x43	1	ASCII
C	грецький	0x39c	01	файли
C	кельтський	0x216d	10	Unicode

5

Алгоритми використання пробілів

1. Зміна кількості пробілів між словами:
один пробіл – "0";
два пробіли – "1".
2. Дописування пробілу перед символом CR/LF:
пробіл відсутній – "0";
пробіл присутній – "1".
3. Дописування зашифрованої частини повідомлення "невидимими" символами перед символом CR/LF.

6

Назва символу	Код	Помітний	.txt	.doc	.pdf	.eml
пробіл	0x0020	+	+	+	+	+
нерозривний пробіл	0x00A0	+	-	+	+	-
пробіл з кельтського алфавіту	0x1680	-	+	+	+	+
En довгий пробіл	0x2000	-	+	+	+	+
Em довгий пробіл	0x2001	-	+	+	+	+
En пробіл	0x2002	-	+	+	-	+
Em пробіл	0x2003	-	+	+	-	+
пробіл ширини 1/3 em	0x2004	+	+	+	+	+
пробіл ширини 1/4 em	0x2005	-	+	-	-	-
пробіл ширини 1/6 em	0x2006	-	+	+	+	+
пробіл завширшки з волосину	0x2007	-	+	+	+	+
синтаксичний пробіл	0x2008	+	+	+	+	+
тонкий пробіл	0x2009	+	+	+	+	+
надтонкий пробіл	0x200A	-	+	+	+	+
тонкий нерозривний пробіл	0x202F	+	+	+	+	+
середній математичний пробіл	0x205F	+	+	+	+	+
ідеографічний пробіл	0x3000	-	+	+	-	+

em й en – ширина й висота символу відносно поточного розміру шрифту, відповідно. 7

Програми-аналоги

Вимоги до наявних стеганографічних сервісів для здійснення порівняння:
- безкоштовність;
- відсутність реєстрації для доступу;
- багаторазовість використання.

Вибрані стеганографічні програми й сервіси:
1. Steganographic Nature of Blankspace (SNOW): darkside.com.au/snow
2. StegZero: stegzero.com

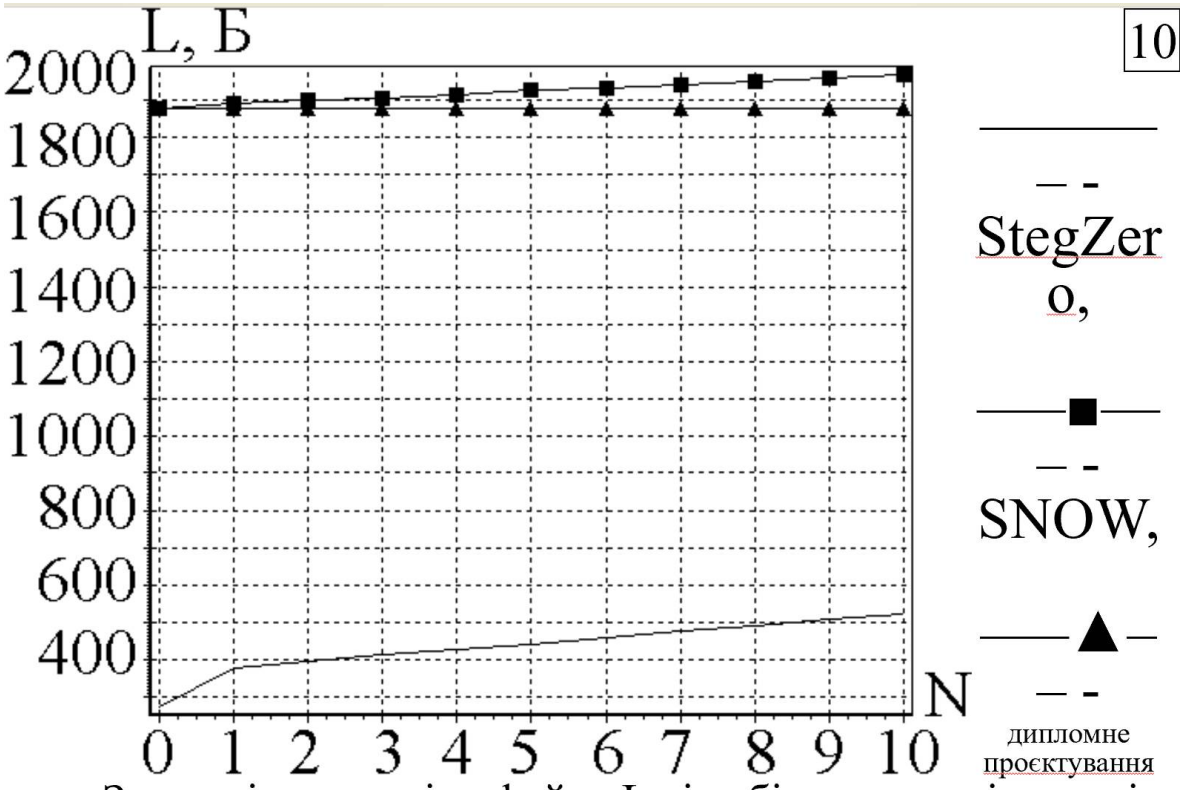
8

Вимоги до реалізації алгоритму однорозрядного кодування:

1. Спосіб кодування - використання роздільників між словами:
- пробіл ("0");
- символ табуляції ("1").
2. Забезпечити сумісність з файлом-контейнером текстового редактора зі складу:
- Microsoft Office;
- Open Office;
- WPS Office;
- Libre Office,

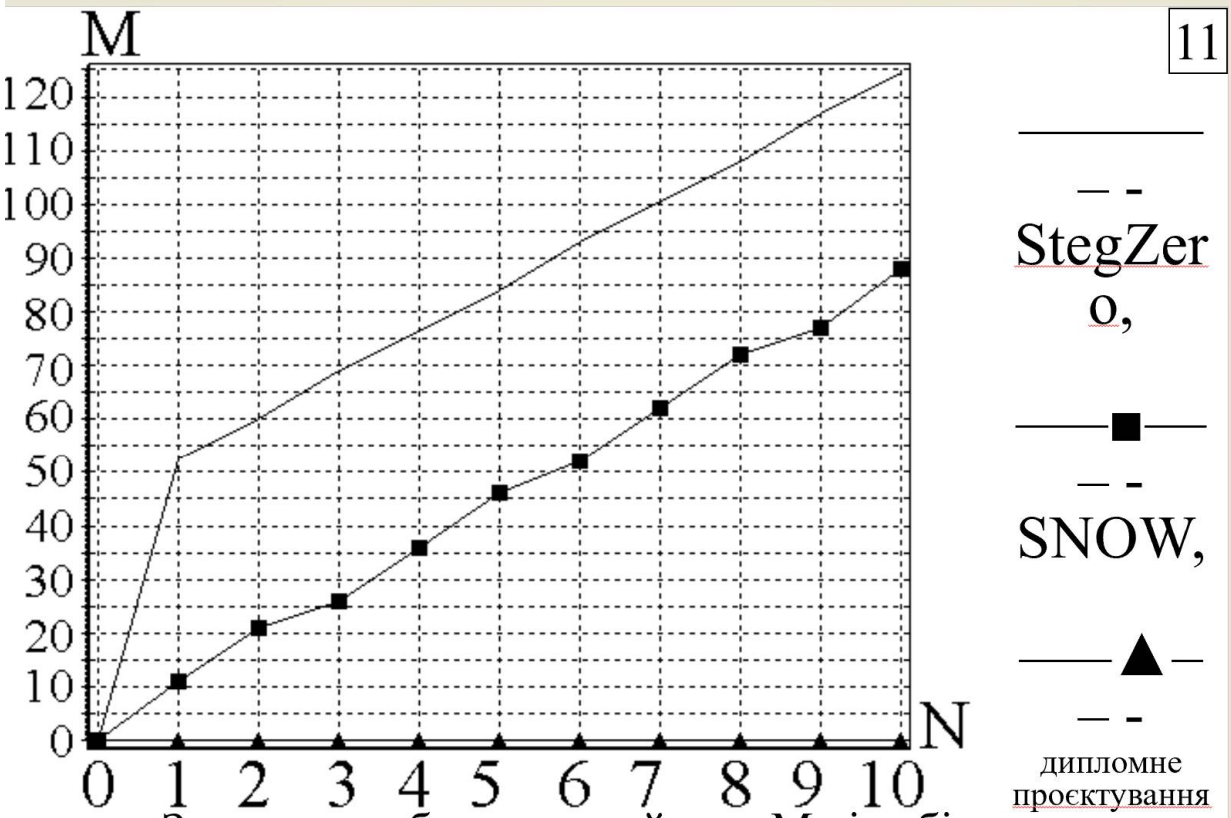
а також за перетворення на файли формату pdf.

9

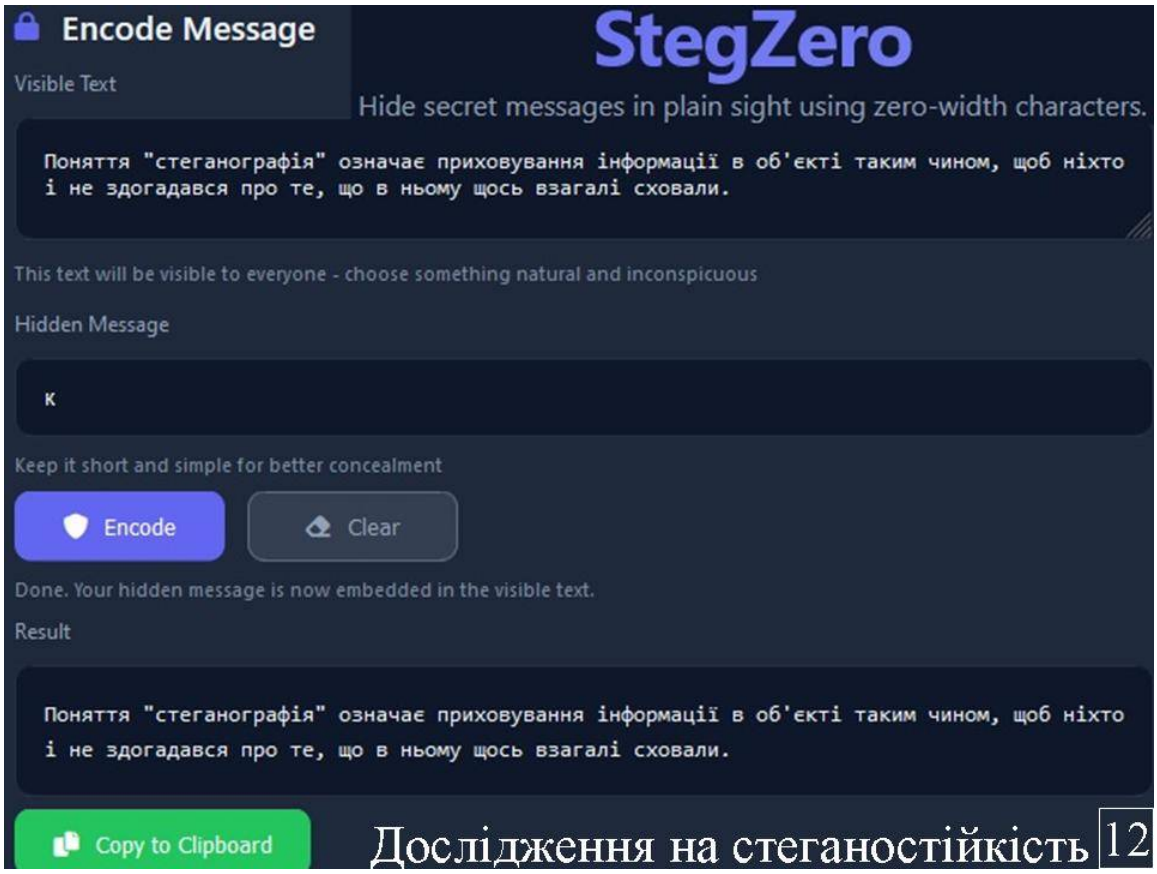


10

Залежність розміру файла L від збільшення кількості символів інформаційного повідомлення N

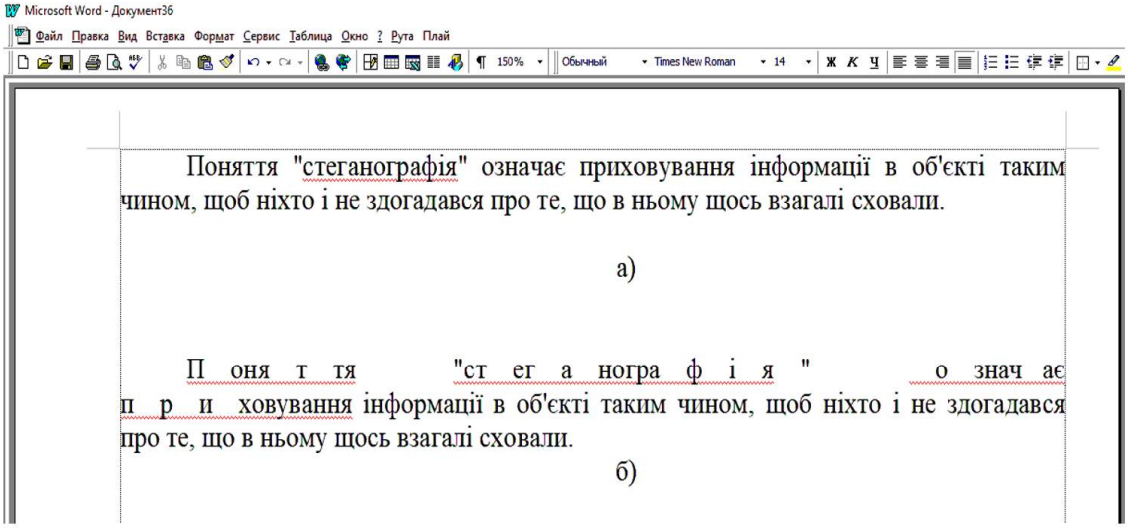


Зростання обсягу контейнера M від збільшення кількості символів інф.повідомлення N



Дослідження на стеганостійкість

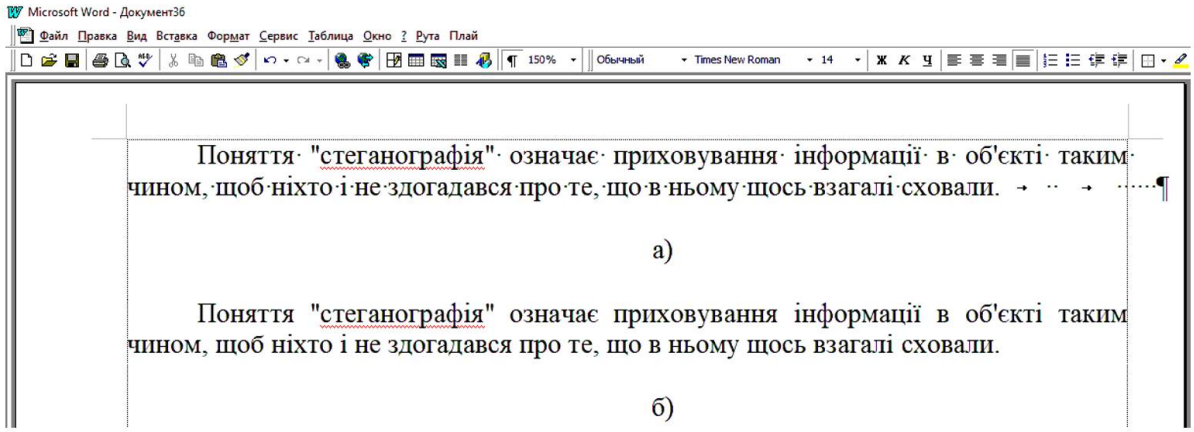
Дослідження на стеганостійкість



Зовнішній вигляд речення у редакторі Word до (а) й після (б) модифікації StegZero

13

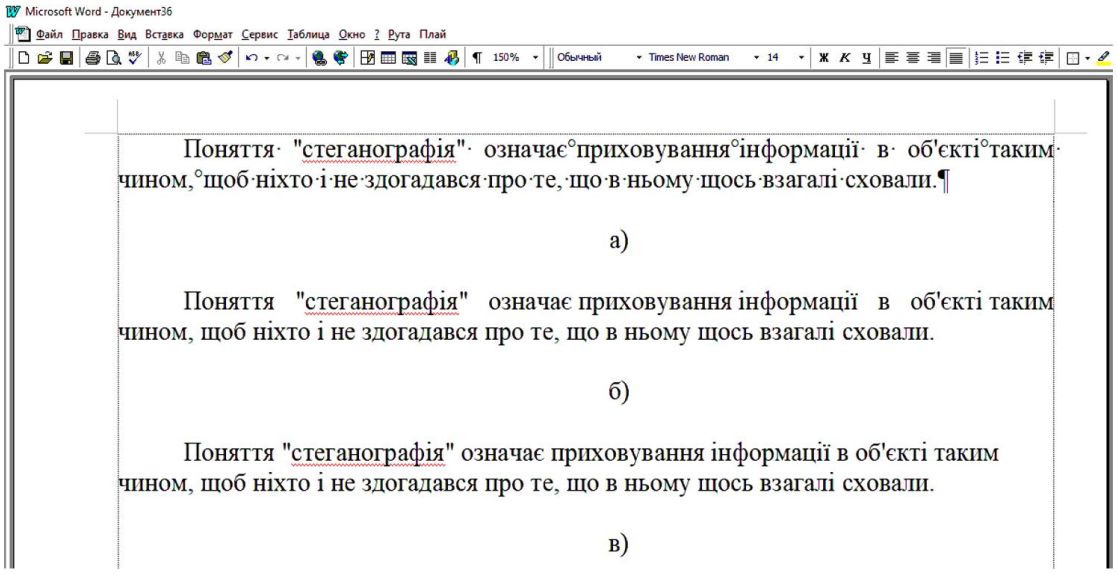
Дослідження на стеганостійкість



Зовнішній вигляд речення у редакторі Word після модифікації SNOW з увімкненими недрукованими символами (а) й вимкненими (б)

14

Дослідження на стеганостійкість



Зовнішній вигляд речення у редакторі Word після модифікації розробленим алгоритмом з увімкненими недрукованими символами (а) й вимкненими: з вирівнюванням за шириною (б) й з вирівнюванням за лівим краєм (в)

15

Табл. 1..3 - Імовірності зустрічі комбінацій серед усіх літер алфавіту

Комбінація	Імовірність зустрічі
01b	26%
10b	26%
11b	27%
00b	21%

Комбінація	Імовірність
001b	12%
010b	9%
011b	17%
100b	12%
101b	14%
110b	18%
111b	9%
000b	7%

Комбінація	Імовірність
0001b	5%
0010b	3%
0011b	8%
0100b	4%
0101b	5%
0110b	11%
0111b	7%
1000b	5%
1001b	7%
1010b	5%
1011b	9%
1100b	8%
1101b	10%
1110b	7%
1111b	2%
0000b	2%

16

Багаторозрядний алгоритм шифрування

1. Усе інформаційне повідомлення подається неперервним двійковим кодом.
2. Якщо N молодших розрядів символу лівіше від пробілу (опускаючи знаки пунктуації) у файлі-контейнері збігається з групою N послідовних чергових розрядів інформаційного повідомлення, то вона кодується символом табуляції. Перейти до п. 2.
3. Якщо молодший розряд символу лівіше від пробілу (опускаючи знаки пунктуації) у файлі-контейнері не збігається із черговим розрядом інформаційного повідомлення, то він кодується парою пробіл або нерозривний пробіл. Перейти до п. 2. 17

Висновки

1. Розроблено модифікацію алгоритму подвійного пробілу для стеганографічного захисту інформації. Рекомендовано цей алгоритм для застосування у документах форматів doc й pdf.
2. Виконано порівняння трьох алгоритмів стеганографічної модифікації контейнера.
3. Зроблено висновок про низьку стеганографічну стійкість алгоритму StegZero для документів у форматі doc й pdf. 18