МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

# МЕТОДИЧНІ ВКАЗІВКИ

до практичних занять з дисципліни

# “СТЕГАНОГРАФІЧНІ СИСТЕМИ”

для студентів усіх форм навчання
спеціальності 125 «Кібербезпека»,
спеціалізації «Безпека інформаційних і комунікаційних систем»

Електронне видання

ЗАТВЕРДЖЕНО
кафедрою БІТ
Протокол № 1 від 29.08.2017

ХАРКІВ 2017

Методичні вказівки до практичних занять з дисципліни “Стеганографічні системи”. Для студентів усіх форм навчання спеціальності 125 «Кібербезпека», спеціалізація «Безпека інформаційних і комунікаційних систем»[Електронне видання] /Упорядник О.І. Федюшин. - Харків: ХНУРЕ, 2017. - 34 с.

Упорядник О.І. Федюшин

Рецензент О.Качко, професор каф. ПЗЕОМ

ЗМІСТ

# ПРАКТИЧНЕ ЗАНЯТТЯ №1
# МАТЕМАТИЧНА МОДЕЛЬ ТА СТРУКТУРНА СХЕМА СТЕГАНОСИСТЕМИ. КРИТЕРІЇ ТА ПОКАЗНИКИ ЕФЕКТИВНОСТІ СТЕГАНОСИСТЕМ.

**1.1 Мета заняття:** вивчити структуру та способи опису математичних моделей стеганографічних перетворень та абстрактне визначення стеганографічних систем; оцінити ефективність функціонування та використання різних моделей стеганографічних систем.

## 1.2 Методичні вказівки з організації самостійної роботи студентів

Сукупність засобів і методів, які використовуються для формування прихованого (непомітного) каналу передачі формують стеганографічну систему або стеганосистему. Основною метою стеганографічного процесу є не шифрування даних, а гарантування того, що вбудовані дані залишаться непоміченими, неушкодженими, та підлягатимуть відновленню.

Існують різні види моделей стеганографічних систем: інформаційно-теоретична модель, теоретико-складна модель, теоретико-ігрова модель, ігрова модель стійкості до атаки активного супротивника, біохімічні моделі стеганографії. Найбільш коректною й у повному ступені, що відображає реальні процеси в стеганографічних системах є інформаційна модель.

### Модель системи стеганографічного приховання даних

Для побудови моделі системи стеганографічного приховання даних вводяться наступні поняття:

1. Стеганографічне поле $SF$ – простір стеганографічного каналу, його об'єкти, а також методи вбудовування й виявлення, тобто

$$SF = \left(SC, C, M, K, \hat{C}, E, D\right), \tag{1.1}$$

де об'єктами стеганографічного поля є:

$c$ – контейнер, $c \in C$ – безлічі всіх контейнерів;

$m$ – повідомлення, $m \in M$ – безлічі всіх повідомлень;

$k$ – ключ, $k \in K$ – безлічі всіх ключів;

$\hat{c}$ – заповнений або модифікований контейнер, $\hat{c} \in \hat{C}$ – безлічі всіх заповнених контейнерів.

2. Метод вбудовування $E$ – набір інструкцій, що виконуються над стеганографічним контейнером для вбудовування повідомлень і одержання модифікованого контейнера:

$$E : C \times M \times K \to \hat{C},\ \hat{c} = E(c, m, k). \tag{1.2}$$

3. Метод виявлення $D$ – набір інструкцій здійснюваних над модифікованим контейнером для виявлення й виймання повідомлень:

$$D : \hat{C} \times K \to M,\ m = D(\hat{c}, k). \tag{1.3}$$

4. Простір стеганографічного каналу $SC$ – просторова, і/або часова, і/або частотна область мультимедійних даних, придатна для стеганографічної передачі повідомлень:

$$\begin{aligned} &F : C \to SC; \\ &SC \subset C. \end{aligned} \tag{1.4}$$

5. Стеганографічна система або стеганосистема $SS$ – сукупність засобів і методів, що здійснюють дії над об'єктами стеганографічного поля в межах простору стеганографічного каналу за допомогою методів вбудовування або виявлення:

$$\begin{aligned} &SS(SC, E) : C \times K \times M \to \hat{C},\ \hat{c} = SS_{SC,E}(c, k, m); \\ &SS(SC, D) : \hat{C} \times K \to M,\ m = SS_{SC,D}(\hat{c}, k). \end{aligned} \tag{1.5}$$

6. *Стеганографічна* модифікація – модифікація контейнера на підставі алгоритму вбудовування:

$$\hat{c} = E(c, m, k). \tag{1.6}$$

Стеганографічний процес можна розбити на 3 етапи [64]. На першому етапі здійснюється вибір об'єктів стеганографічного поля $c \in C$, $m \in M$, $k \in K$. У якості приховуваних даних може використовуватися будь-яка інформація: текст, аудіо-файл, зображення й т.п. Дану інформацію прийнято називати приховуваним повідомленням або просто повідомленням.Файлом-контейнером називається файл, призначений для приховання в ньому конфіденційної інформації, причому $M \subset C$. Вибір контейнера впливає на надійність стеганосистеми й на можливість виявлення факту передачі приховуваного повідомлення. Розмір контейнера безпосередньо впливає на пропускну здатність стеганографічного каналу передачі даних.

На другому етапі вибирається метод вбудовування/виявлення $E, D$. На третьому етапі здійснюється генерація стеганоключа. Стеганоключ або просто ключ – деяка секретна інформація, відома тільки законному користувачеві, необхідна для приховання повідомлення. Залежно від рівня захисту (наприклад,

вбудовування попереднє зашифрованого повідомлення) у стеганосистемі може бути один або кілька ключів. Ключ може бути представлений псевдовипадковою послідовністю біт, породжуваною генератором певним, що задовольняє вимогам (криптографічно безпечний генератор). Розрізняють два типи стеганосистем:

- з відкритим ключем;
- з секретним ключем.

У системах з відкритим ключем використовуються два ключі $k_o \in K_o$ й $k_{os} \in K_{os}$, незалежні друг від друга. Один із ключів $k_o \in K_o$ є несекретним, тобто може передаватися вільно по незахищеному каналу зв'язку, другий $k_{os} \in K_{os}$, є секретним, і не може бути отриманий за допомогою обчислень з ключа $k_o \in K_o$ .

Одним із прикладів використання систем з відкритим ключем може бути наступний набір ключів: $k_o$ = “Дівоча прізвище матері?”, $k_{os}$ = “Іванова”. Стеганографічна система в цьому випадку буде здійснювати дію над об'єктами у відповідність із ключем $k_{os}$:

$$SS(SC,E): C \times K_{os} \times M \to \hat{C},\ \hat{c} = SS_{SC,E}(c, k_{os}, m). \qquad (1.7)$$

У стеганосистемі із секретним ключем використовується один секретний ключ $k_s \in K_s$, який повинен бути визначений або до початку обміну приховуваними повідомленнями, або переданий по захищеному каналу. Стеганографічна система буде здійснювати дію над об'єктами у відповідності із ключем $k_s \in K_s$:

$$SS(SC,E): C \times K_s \times M \to \hat{C},\ \hat{c} = SS_{SC,E}(c, k_s, m). \qquad (1.8)$$

Таким чином, стеганографічний процес можна представити в наступному вигляді (рисунок 1.1):

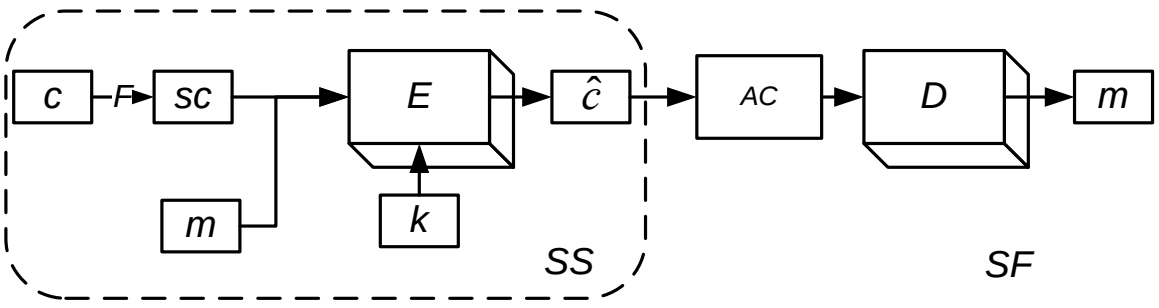



Рисунок 1.1 – Узагальнена схема процесу стеганографічного

прихования даних

Під каналом атак *AC* мається на увазі складова стеганографічного поля, де здійснюються неправомірні дії порушника. Порушником називають особу, що робить протиправні дії, спрямовані на виявлення, виймання або руйнування повідомлення. Під стійкістю різних стеганосистем або стеганостійкістю розуміється їхня здатність приховувати від кваліфікованого порушника факт існування стеганографічного каналу, здатність протистояти спробам порушника зруйнувати, спотворити, вилучити потай передані повідомлення, а також здатність підтвердити або спростувати дійсність потай переданої інформації. Стеганостійкість оцінюється процентним відношенням кількості цифрових файлів у яких порушникові вдалося виявити факт наявності вбудованих повідомлень або непомітно перешкодити їхній потайливій передачі до загального числа досліджуваних файлів.

Під пропускною здатністю стеганосистеми розуміється відношення максимальне можливого обсягу вбудованого повідомлення до обсягу файлу-контейнера при дотриманні вимог стеганостійкості.

**Методи модифікації зображень у стеганографічних системах прихования даних.**

У цей час у світі існує ряд алгоритмів непомітного вбудовування інформації в зображення. У більшості вони засновані на декомпозиції контейнера. Методи прихования інформації в зображеннях, використовувані в стеганоалгоритмах діляться на:

1. Методи, що здійснюють прихования в просторовій області контейнера-зображення. Стеганографічній модифікації підлягають найменш значимі, надлишкові, з погляду зорової системи людини, біти контейнера. Наприклад, LSB-метод (модифікації піддаються значення компонентів колірної моделі зображення), метод квантування зображення, метод блокового прихования, метод заміни палітри, метод псевдовипадкового інтервалу [22, 23].

2. Методи, що здійснюють прихования в частотній області контейнера-зображення. Стеганографічній модифікації піддаються, наприклад, коефіцієнти дискретного косинусного перетворення (ДКП), вейвлет-перетворення й ін. Методи, що використовують вейвлет-перетворення й ДКП, одержали найбільше поширення, тому що вони проявляють високу стеганостійкість до атак на стиснення на відміну від просторових методів. Це пояснюється

використанням при стеганографічній модифікації аналогічного математичного перетворення, що й при стисненні зображень. У форматі jpeg використовується ДКП, в jpeg2000 – вейвлет-перетворення. Однак алгоритм приховання, що використовує вейвлет-перетворення, не є стеганостійким до алгоритму стиснення, що використовує ДКП. Ефективність застосування даних методів пояснюється тим, що вони моделюють процес обробки зображень у системі зору людину й відокремлюють «значимі» деталі від «незначущих».

3. Методи, що не піддають контейнер-зображення модифікаціям. Особливістю даних методів є відсутність модифікацій контейнера й наявність унікальної для кожного повідомлення ключової таблиці, яка передається разом з порожнім контейнером.

Методи, що не піддають контейнер модифікаціям, повністю задовольняють вимогам стеганостійкості. Основною складової стеганографічної системи є унікальна ключова схема, у відповідності із якою відбувається пошук подібних областей повідомлення, що вбудовується, і контейнера. Добування повідомлення здійснюється при обов'язковій наявності ключової схеми. Основним недоліком даних методів є неможливість використання незмінного ключа, що зобов'язує здійснювати постійну передачу ключових даних за допомогою іншого стеганографічного або криптографічного каналу.

**Оцінка ступеня придатності зображень для стеганографічної модифікації .**

Стеганографічній модифікації можуть піддаватися як просторові, так і частотні параметри контейнера-зображення. Для оцінки можливих меж стеганографічної модифікації параметрів зображень, доцільно провести класифікацію контейнерів-зображень.

Кожне зображення має унікальні властивості, які можна покласти в основу їх поділу на класи. При візуальній роботі із зображеннями найбільш широко використовується наступна класифікація.

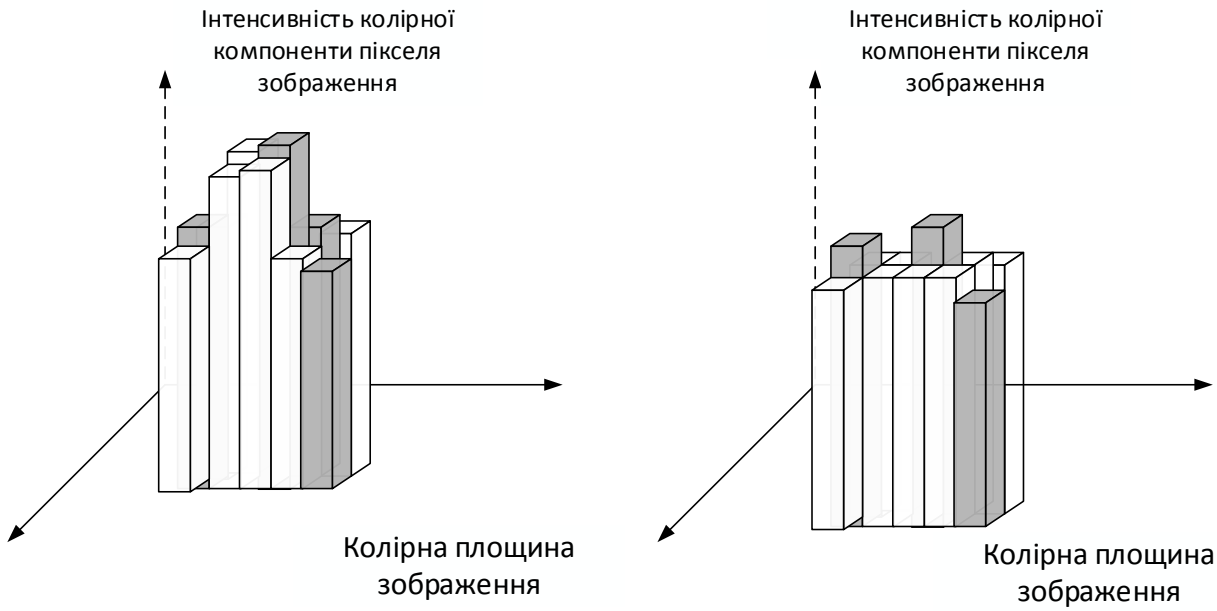
1. Зображення з невеликою кількістю кольорів (4-16) і великими областями, заповненими одним кольором.
2. Зображення, побудовані на комп'ютері, із плавними переходами кольорів.
3. Фотореалістичні зображення.
4. Фотореалістичні зображення з накладенням ділової графіки.

5. Картографічні зображення.
6. Космічні зображення.

Дана класифікація базується не на фізичних параметрах зображень, що унеможливлює її застосування як об'єктивну оцінку ступеня придатності зображень для стеганографічної модифікації.

Для забезпечення непомітності факту приховання даних у контейнерах-зображеннях необхідно вибирати зображення, що містять області з різкими переходами кольорів, границі об'єктів, тому що в них можна в більшому ступені сховати невеликі зміни інтенсивностей колірних компонентів пікселя контейнера стосовно сусідніх пікселів. Модифіковані пікселі на рисунку виділені сірим кольором. Видно, що навіть незначна зміна інтенсивності якого-небудь колірного компонента окремих пікселів, в областях, заповнених одним кольором (рисунок 1.2,б), збільшує можливість візуального або комп'ютерного детектування факту стеганографічного приховання даних.

Аналіз умісту контейнерів-зображень доцільно проводити за допомогою математичного перетворення, що дозволяє виділити його частотні параметри й оцінити внесок окремих частот до складу зображення.




(а) (б)

Рисунок 1.2 – Колірні площини модифікованих областей зображення: (а) область різкого переходу кольорів, (б) область, заповнена одним кольором

Одним з таких перетворень є дискретне косинусне перетворення, застосовуване до зображень за допомогою вікон розмірів $n \times n$ пікселів, де $n$ звичайно вважається рівним 8. Дискретне косинусне перетворення дозволяє одержати інформацію про різкі й плавні границі кольорів зображення, областях, заповнених одним кольором або із градієнтною зміною кольору й ін. Оцінку ступеня придатності зображення для стеганографічної модифікації доцільно проводити у два етапи.

На першому етапі зображення діляться на класи згідно з відносною вагою їх просторових частот. Позитивні результати отримані при використанні спектральної класифікації зображень, що дозволяє провести їх розбиття на 8 класів. Класи з 1-ого по 3-ий описують зображення з найбільшою відносною вагою низьких частот; класи з 4-ого по 7-ой розмежовують зображення по спектральних складових, зосереджених в областях близьких до низькочастотного й/або високочастотному діапазонам; восьмий клас відокремлює зображення, що мають рівномірний спектр у межах усього розглянутого діапазону.

На другому етапі оцінюється пропускна здатність зображення-контейнера шляхом виключення непридатних для модифікації областей, наприклад, заповнених одним кольором або градієнтною зміною кольору. Виявити такі області на зображенні можливо при аналізі його спектрального складу (наявність границь на зображенні призводить до збільшення внеску середніх і високих частот), так і при оцінці зміни інтенсивності пікселя стосовно сусідніх пікселів.

**Класифікація даних, що вбудовуються.**

Непомітність факту приховання даних і забезпечення високої стеганостійкості досягається не тільки оптимальним вибором контейнера. Величезну роль представляють дослідження структур, даних що вбудовуються.

Аналіз статистичних параметрів файлу-контейнера, поведінки молодших значущих біт до й після модифікації, дозволив виявити різний вплив даних, що вбудовуються. По характеру впливу, що здійснюється на параметри контейнера, безліч повідомлень, що вбудовуються $M$ , можна розділити на:

1. Файли повідомлень, що підлягають процедурі стиснення $m_1$ (наприклад, файли у форматах JPEG, RAR, MP3 і ін.);

2. Файли повідомлень, несхильні до процедури стиснення $m_2$ (наприклад, файли у форматах BMP, DOC, WAV, ICO і ін.).

Докладний розгляд повідомлень даних типів з використанням шістнадцяткового редактора дозволило визначити відмінності в структурі файлів (рисунок 1.3).

Повідомлення, несхильне до стиснення (рисунок 1.3,б), на відміну від повідомлення, що піддається стисненню (рисунок 1.3,а), має ланцюжки біт однакових або близьких значень. Це впливає на характеристики контейнера при стегановбудовуванні.

```
000005C0: BA 20 63 0C CD 40 F0 42 | C4 B1 21 08 49 1E 4C 43
000005D0: 48 8E 3C 69 EB 7A 41 9A | F8 63 70 1B 04 D0 9A 68
000005E0: 20 C8 42 AA 8E C8 32 1A | F5 E9 08 D9 A4 70 8E 1F
000005F0: 62 D7 C4 1B D2 2A E0 12 | 3A 71 86 E5 A9 38 E0 89
```

(а)

```
000A1C20: 07 00 00 00 4C 00 00 00 | 00 00 00 00 00 00 00 00
000A1C30: 00 00 00 00 00 00 FF FF | FF FF FF 00 00 00 00 00
000A1C40: 3C 04 3E 04 34 04 35 04 | 3B 04 4C 04 20 00 00 00
000A1C50: 0A 00 00 00 08 00 00 00 | 08 00 00 00 07 00 00 00
```

(б)

Рисунок 1.3 – Фрагменти повідомлень, що вбудовуються, у шістнадцятковому кодуванні: (а) повідомлення, що піддається стисненню, (б) повідомлення, що не піддається стисненню

Кількість поява значень відліків незаповненого контейнера розподілена за законом, близьким до нормального розподілу. Вбудовування повідомлень першого типу викликає модифікації відліків контейнера, які незначно впливають на загальний розподіл у силу відсутності у ланцюжків даних однакових значень у структурі файлу.

При вбудовуванні повідомлень другого типу відбуваються модифікації контейнера, що змінюють характер розподілу його відліків і поведінку молодших значущих біт.

Таким чином, для підвищення стеганостійкості й непомітності модифікації повідомлення, що вбудовується, по своїх статистичних параметрах повинні узгоджуватися з файлами-контейнерами.

**Класифікація стеганографічних схем вбудовування повідомлень.**

В основу вибору алгоритму вбудовування в більшості випадків покладені результати аналізу стійкості стеганографічного каналу. Одним з напрямків, що

дозволяють підвищити стеганостійкость, є використання ключової схеми при вбудовуванні повідомлення, причому різним ключовим стеганографічним схемам властиві різні рівні стеганостійкости. Відомо, що при підвищенні стеганостійкості пропускна здатність каналу зменшується, тому завдання вибору схеми вбудовування при забезпеченні оптимальних параметрів стеганосистеми в цілому не є тривіальною.

Вбудовування повідомлення в стеганографічній системі може здійснюватися як з використанням ключа, так і без його використання. Для підвищення стеганостійкості системи ключ можна використовувати в якості верифікаційного інструмента. Він може впливати так само на розподіл біт повідомлення в межах контейнера й на порядок формування послідовності біт повідомлення, що вбудовуються. Можна виділити чотири стеганографічні схеми вбудовування, що забезпечують різні рівні захищеності або стійкості стеганографічного каналу передачі приховуваних повідомлень.

Перший (найнижчий) рівень захисту забезпечується тільки вибором алгоритму вбудовування $E$ (1.2). Це може бути алгоритм заміни молодших значущих біт контейнера, або алгоритми модифікації частотних або просторово-часових характеристик контейнера. Перший рівень захисту присутній в будь-якому стеганографічному каналі передачі повідомлень. Стеганографічна система в цьому випадку може бути представлена, як показано на рисунку 1.4.

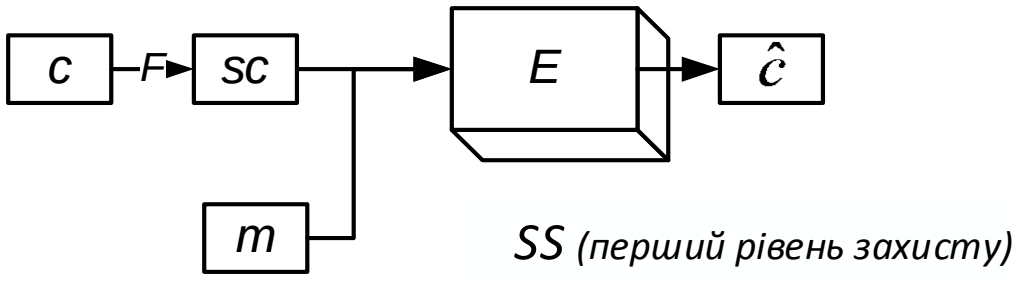



Рисунок 1.4 – Стеганографічна система з першим рівнем захисту

Наведена схема знайшла своє застосування в такому стеганографічному програмному продукті, як, наприклад, Fortknox 3.55.

Другий рівень захисту стеганосистеми вимагає використання ключових стеганографічних схем, що припускають запис немодифікованого або модифікованого пароля в початок або в кінець повідомлення, розподіл парольного підпису по всій довжині стеганографічного каналу. Такі ключові

схеми не впливають на розподіл повідомлення по контейнеру й не піддають повідомлення попередній обробці згідно з обраним ключем (рисунок 1.5).

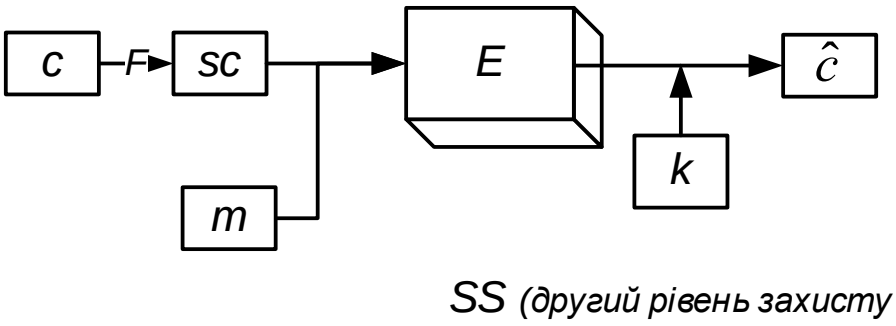



Рисунок 1.5 – Стеганографічна система із другим рівнем захисту

Подібні стеганографічні системи знаходять застосування в таких задачах, як, наприклад, впровадження цифрового підпису для доказу авторських прав. Швидкість обробки контейнера при цьому не змінюється. Дана ключова схема знайшла застосування в наступних стеганографічних програмних продуктах: Data Stash , C1oak 7.0 , Steganography 1.50 і Data Stealth 1.0 .

Більш захищеними є стеганографічні канали передачі даних із ключовими схемами вбудовування інформації, у яких по ключу здійснюється розподіл повідомлення по контейнеру (рисунок 1.6,а). Відповідно, швидкість обробки контейнера буде нижче, чим у випадку застосування першої й другої ключових схем. Процес стеганокодування по третій ключовій схемі представляється у вигляді наступного алгоритму:

1. Визначення вхідних параметрів стеганосистеми.

Вхідними параметрами для функції розподілу повідомлення по контейнеру будуть $m \in M$ , $c \in C$ , $k \in K$ .

2. Визначення $L$ – мінімального числа відліків контейнера, необхідного для впровадження одного відліку повідомлення.

Наприклад, для вбудовування 8-бітного відліку повідомлення при кодуванні молодших біт 8-бітного аудіо-контейнера необхідно 8 байт цього контейнера. Значення $L$ в загальному випадку визначається алгоритмом вбудовування.

3. Визначення $P$ – кроку розподілу повідомлення по контейнеру.

На підставі введеного відправником ключа вбудовування $k$ , у загальному випадку, що представляє собою рядок пароля (набір символів), визначається крок $P$ розподілу повідомлення $m$ по контейнеру $c$ . Одним з варіантів

перетворення рядка символів у число є пошук відповідності в таблиці ASCII кодів. Обчислення кроку розподілу повідомлення по контейнеру включає наступні операції:

- Додавання ASCII кодів усіх символів рядка пароля;

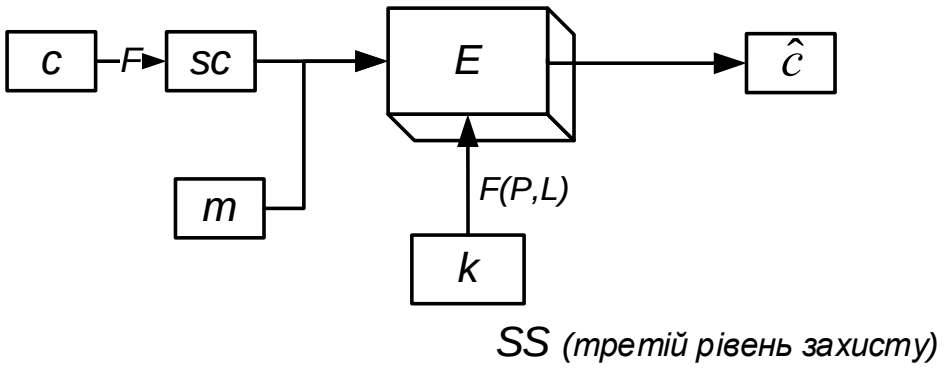
– Розподіл суми на задане відправником число $S_p$;

– Використання залишку від розподілу суми на число $S_p$ як крок розподілу $P$.

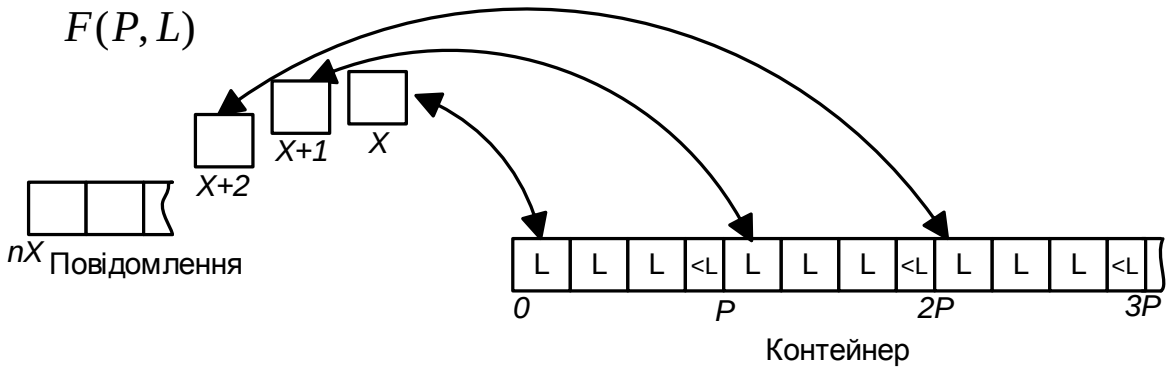
Таким чином, відправник перетворить ключ у число $P = P(k, S_p)$.

Однією з обов'язкових умов можливості розподілу повідомлення по контейнеру відповідно до ключа є дотримання умови

$$P \geq L. \tag{1.9}$$




(а)




(б)

Рисунок 1.6 – Стеганографічна система із третім рівнем захисту:
(а) узагальнена схема; (б) функція розподілу повідомлення по контейнеру $F(P, L)$

4. Завдання функції розподілу $F(P,L)$ повідомлення по контейнеру.

Відповідно до обчисленого кроку $P$ контейнер розбивається на блоки – своєрідні «ящики» для зберігання відліку повідомлення. Розмір кожного блоку дорівнює кроку $P$. Останній блок по величині може бути менше $P$, якщо ємність контейнера не кратна $P$. Кожний блок контейнера містить ряд зон вбудовування ємністю $L$ кожна, причому остання зона в блоці може бути менш $L$ і тому вона не підлягає вбудовуванню.

Розподілу біт повідомлення по контейнеру здійснюється за принципом наскрізного проходу один по одному. Перший відлік $x$ повідомлення заповнює перший блок контейнера від позиції 0 у першій зоні вбудовування до позиції $L$ (рисунок 1.6,б) наступний відлік $x+1$ повідомлення, аналогічним образом записується в другий блок контейнера і так далі. Після заповнення повідомленням усіх перших зон вбудовування блоків контейнера, у тому ж порядку заповнюються наступні зони блоків контейнера, починаючи з першого блоку. Відліки повідомлення слідують один за одним: $x$, $x+1$, $x+2$ … $nx$ .

Таким чином, функція розподілу повідомлення по контейнеру представляється в такий спосіб:

$$F(P,L) = cycle * L + step * P, \qquad (1.10)$$

де $step$ – номер кроку вбудовування, $cycle$ – номер поточної зони $L$.

При такому вбудовуванні повідомлення необхідно враховувати номер поточного проходу контейнера, щоб не допустити перетинань і вбудовування у вже заповнені гнізда, а також номер блоку повідомлення, щоб уникнути перетинань і зберегти можливість добування повідомлення.

5. Вбудовування повідомлення.

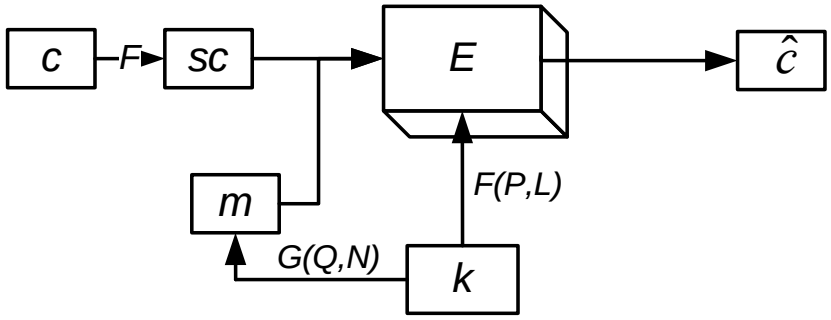
Повідомлення вбудовується в контейнер відповідно до алгоритму вбудовування $E$ й порядком, визначеним функцією розподілу $F(P,L)$. Однієї з особливостей даного алгоритму є те, що якщо розмір зони $L$ кратний розміру блоку $P$, а розмір блоку $P$ кратний розміру контейнера, пропускна здатність такого каналу не зменшується. Відбувається зміна впорядкування повідомлення по контейнеру відповідно до функції розподілу, однак усі його області заповнюються рівномірно й не залишається порожніх ділянок.

Відмінність четвертої ключової схеми від третьої полягає в тому, що в стеганосистемі використовуються дві функції розподілу повідомлення по контейнеру. Перша відповідає за порядок вибору відліків повідомлення

відповідно до деякої функції $G(Q,N)$, а друга функція $F(P,L)$ відповідає за вибір позиції в контейнері для вбудовування відліку повідомлення.

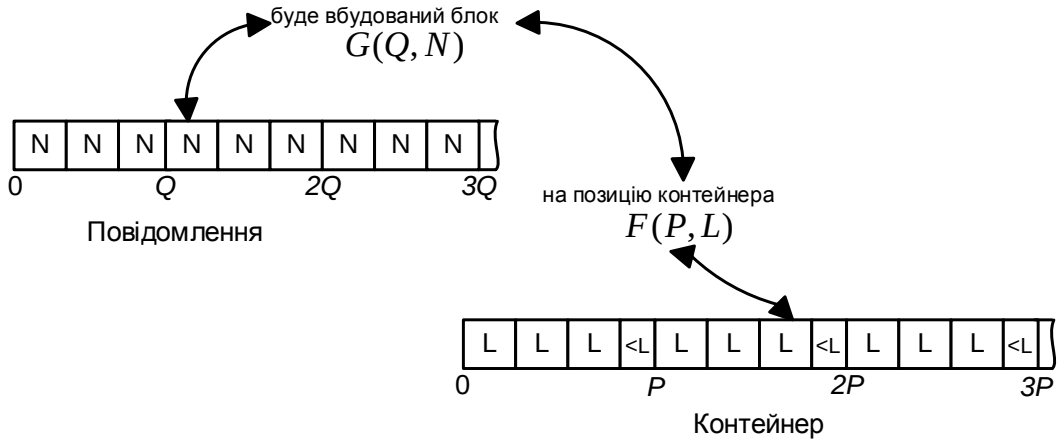
Стеганографічна система із четвертим рівнем захисту представлено на рисунку 1.7.

Процес приховання даних відповідно до четвертої ключової схеми представляється у вигляді алгоритму:




*SS (четвертий рівень захисту)*

(а)




(б)

Рисунок 1.7 – Стеганографічна система із четвертим рівнем захисту:
(а) узагальнена схема; (б) функція $G(Q,N)$

1.-4. Дані кроки алгоритму ідентичні тим, що використовуються в третій ключовій схемі.

5. Функцією вибору відліків повідомлення є функція $G(Q,N)$, по суті аналогічна функції $F(P,L)$. Повідомлення, що вбудовується, розбивається на блоки *розміром Q*. Величина $Q$ відповідно до деякої функції перетворення $Q = Q(k, S_Q)$ обчислюється по рядкові символів, використовуваної в якості

секретного ключа або пароля. Кожний блок $Q$ по величині повинен задовольняти умові:

$$Q \geq N, \tag{1.11}$$

де $N$ – розмір відліку повідомлення в бітах.

Мінімальний розмір $N$ становить 1 біт, $Q$ завжди ділиться на $N$ без залишку, для вбудовування в контейнер кожного біта повідомлення.

У якості функції вибору біт повідомлення використовується функція

$$G(Q,N) = cycle * N + step * Q \ , \tag{1.12}$$

де $step$ – номер блоку відліку, $cycle$ – номер поточної області $N$.

6. Вбудовування проводиться відповідно до функцій $G(Q,N)$ і $F(P,L)$ .

На підставі викладених вище міркувань можна побудувати класифікаційну таблицю ключових стеганографічних схем (таблиця 1.1).

Таблиця 1.1 – Класифікаційна таблиця ключових стеганографічних схем

| Рівень захисту стеганографічного каналу | Наявність стегано-графічного алгоритму | Наявність ключа | Вплив ключа на розподіл біт повідом-лення по контейнеру | Вплив ключа на порядок формування послідовності біт повідомлення, що вбудовуються |
|---|---|---|---|---|
| 1 | + | – | – | – |
| 2 | + | + | – | – |
| 3 | + | + | + | – |
| 4 | + | + | + | + |

### 1.3 Контрольні запитання та завдання:

1. Що включає математична модель стеганографічної системи?
2. Побудуйте структурну схему стеганографічної системи.
3. Охарактеризуйте класифікацію стеганоконтейнерів.
4. Чим відрізняються прихований та відкритий стеганоконтейнери?
5. Чим відрізняються порожній та потоковий контейнери?

6. Назвіть основні вимоги до стеганосистем з секретним ключем.

7. Назвіть основні вимоги до стеганосистем з відкритим ключем.

8. Що таке криптографічна стійкість?

9. Що таке стеганографічний канал?

10. Назвіть основні вимоги до стеганографічного каналу.

11. Чим відрізняються стеганодетектори від стеганодекодерів?

12. У чому полягає афінне перетворення зображення?

13. Які етапи є в алгоритмі вбудовування повідомлення в найпростішому випадку?

14. Охарактеризуйте структурну схему стеганосистеми як системи зв’язку.

15. Охарактеризуйте методи стеганографії.

**Завдання.**

Вивчити теоретичний матеріал щодо функціонування та створення стеганографічних систем різноманітного призначення, ознайомитись з математичними моделями опису стеганографічних систем.

Скласти блок-схеми алгоритмів функціонування ключових стеганографічних схем 1-4 рівнів.

Провести порівняльний аналіз схем, і навести приклади їх практичного використання.

# ПРАКТИЧНЕ ЗАНЯТТЯ №2
# ПРИХОВУВАННЯ ДАНИХ У ТЕКСТОВИХ ДОКУМЕНТАХ

**2.1 Мета заняття:** вивчити та дослідити функціонування основних алгоритмів приховування даних у текстових документах, отримати навички їх використання на практиці для захисту конфіденційних даних.

## 2.2 Методичні вказівки з організації самостійної роботи студентів

Особливо складними з мультимедійних об’єктів для приховування даних з багатьох причин являються текстові файли. Це пов’язано з тим, що текстовий файл може бути оброблений при підготовці на друк, добавляння додаткової букви або знака пунктуації в тексті можуть бути легко розпізнанні випадковим читачем. Така ситуація спричиняє відносний дефіцит у текстовому файлі надлишкової інформації, особливо порівняно з графічними або звуковими файлами. Виділяють три групи текстових методів приховування інформації:

− методи довільного інтервалу;

− синтаксичні методи;

− семантичні методи.

Методи довільного інтервалу в певних випадках показують досить непогані результати. По-перше, зміна кількості пробілів у кінці текстового рядка не викликає істотних змін у значенні фрази або реченні. По-друге, середньостатистичний читач навряд чи помітить незначні модифікації вільного місця сторінки тексту.

До методів даної групи належать:

1. **Метод зміни інтервалів між реченнями**, який дозволяє вбудовувати в текст повідомлення шляхом розміщення одного або двох відступів після кожного символу завершення речення. Даний метод простий в використанні, однак має ряд недоліків: для вбудовування незначної кількості біт потрібен текст значного розміру; залежить від структури текстового контейнера (в деяких текстових контейнерах можуть бути відсутні знаки завершення рядка, деякі текстові редактори можуть автоматично добавляти після крапки відступи).

2. **Метод зміни кількості відступів у кінці текстових рядків**, що полягає в додаванні відступів у кінець кожного текстового рядка. Кількість добавлених відступів залежить від значення вбудованого біта. Два відступи кодують один біт на рядок, чотири відступа – два біти і так далі. Такий підхід

дозволяє істотно збільшити, порівняно з попереднім методом, кількість інформації, яку можна приховати в тексті аналогічного об'єму. Даний метод може бути застосований до будь-якого тексту, при чому зміни у форматі останнього будуть у достатній мірі непомітними. Недоліком даного методу є те, що деякі програми обробки тексту можуть ненавмисно видаляти додатково внесені відступи;

3) **Застосування методу зміни кількості відступів між словами** вирівняного по ширині тексту дозволяє приховувати дані у вільних місцях тексту, вирівняного по ширині. При цьому біти даних вбудовуються шляхом керованого вибору позицій, в яких будуть розміщені додаткові відступи. Один відступ між словами інтерпретується як «0», а два відступи – як «1». В середньому метод дозволяє вбудовувати по кілька біт в один рядок. Недоліком є те, що через обмеження, які накладаються вирівнюванням тексту по ширині, не кожен відступ між словами може використовуватися для вбудовування даних.

**Синтаксичні методи** полягають в зміні пунктуації, структури та стилю тексту. Дані методи слід використовувати з ретельною обачністю, бо зміна пунктуації може призвести до зниження сприйняття тексту, надати протилежного змісту чи привернути увагу цензора.

**Семантичні методи** подібні до синтаксичним. Вони визначають два синоніма, які відповідають значенням приховуваних біт. Наприклад, слово «проте» може бути поставлено у відповідність до «0», а слово «однак» – до «1». Для використання даних методів необхідно наявність таблиці синонімів. Якщо слову відповідає велика кількість синонімів, то можливо одночасно приховувати більшу кількість бітів. Проблемою даних методів для вбудовування біта інформації може перешкоджати особливість значення слова.

Стеганографія, що використовує текстові контейнери, називається текстовою (text steganography). Далі буде розглянуто, яким чином можна застосовувати текстові контейнери для зберігання стега. З автоматичних методів текстової стеганографії в цій роботі згадується тільки один – *форматування*, тобто вирівнювання, *тексту за допомогою пробілів*.

Суть даного методу полягає в розсуненні рядка шляхом збільшення пробілів між словами, коли один пробіл відповідає, наприклад, біту 0, два пробіли – біту 1. Однак пряме його застосування хоча й можливе, але на практиці породжує масу незручностей, зокрема, оформлення тексту стає неохайним, що дозволяє легко запідозрити в ньому наявність стега.

*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.

*Метод хвостових пробілів* припускає дописування наприкінці коротких рядків (менше 225 символів; значення 225 обране досить довільно) від 0 до 15 пробілів, що кодують значення напівбайта.

*Метод знаків однакового накреслення* припускає підміну (бітів 1) або відмову від такої підміни (бітів 0) російського символу латинським того ж накреслення.

*Метод двійкових нулів* є різновидом методу знаків однакового накреслення й припускає або заміну першого в групі із двох або більше внутрішніх пробілів двійковим нулем (бітів 1), або відмову від неї (бітів 0).

Розглянемо декілька з описаних методів більш докладніше, і проведемо їх практичне дослідження з використанням математичного пакету MathCad.

**Метод зміни інтервалу між реченнями.** В основі цього методу лежить така ідея: окремі біти ASCII-кодів (представлених у двійковому форматі) символів конфіденційного повідомлення приховуються у відкритому текстовому файлі після кожного символу кінця речення (це може бути символ крапки у звичайному тексті або символ крапки з комою, наприклад, для програмного коду, написаного мовою C++ тощо) за допомогою одного чи двох пропусків. Одним пропуском стенографічно кодується, наприклад, біт “1”, двома – біт “0”.

Як порожній контейнер розглянемо текст, фрагмент якого наведено на рис. 2.1. Приховуване повідомлення “Ключ”.

Результат вбудовування першого символу повідомлення (“К”), двійковий формат ASCII-коду якого: $D2B(str2vec("К"))^T = (0_{LSB}\ 1\ 0\ 1\ 0\ 0\ 1\ 1_{MSB})$, зображено на рис. 2.2. Для наочності і кращого розуміння алгоритму методу зміни інтервалу між реченнями зафарбованими клітинками ми позначили ті пропуски, у яких вбудовано окремі біти конфіденційного повідомлення. Так, у кожному міченому кольором одинарному символі пропуск сховано одиничний біт повідомлення, а у кожній парі – нульовий біт.

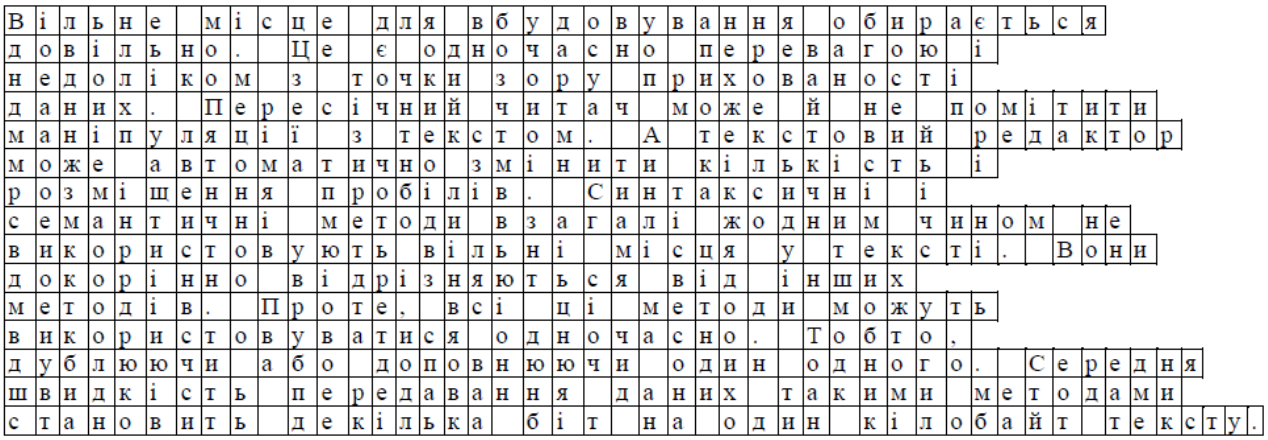


Рисунок 2.1 – Фрагмент текстового контейнера, що використовується для приховування даних

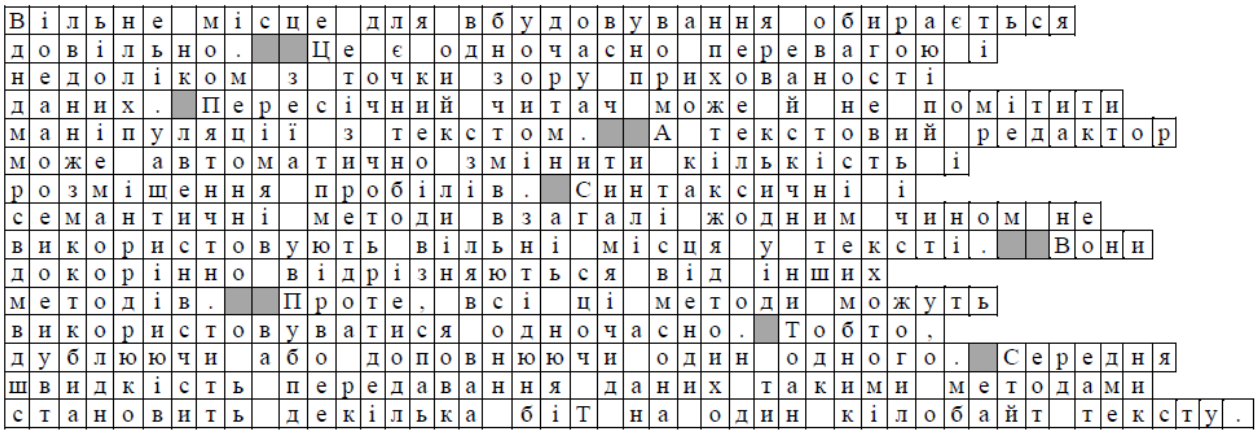


Рисунок 2.2 – Фрагмент заповненого методом зміни інтервалу між реченнями текстового контейнера

Розглянемо недоліки описаного методу. По-перше, він є неефективним, оскільки потребує великого за обсягом тексту порожнього контейнера для вбудовування повідомлення незначного розміру. За умови, що середньостатистичне речення займає 2 рядки по 80 символів кожен, пропускна здатність методу – 0,08%. По-друге, можливість приховування сильно залежить від структури тексту-контейнера (деякі тексти, як наприклад, вільні вірші характеризуються відсутністю стійких узгоджених або однозначних знаків завершення речення). По-третє, деякі з текстових редакторів автоматично додають після знаку кінця речення один-два пропуски (автозавершення), що зводить нанівець процедуру видобування повідомлення.

**Метод хвостових пропусків.** Метод хвостових пропусків полягає у дописуванні в кінець поточного рядка порожнього файлу-контейнера одного символу пропуску у разі вбудовування в нього, наприклад, нульового біта стеганоповідомлення. Одиничний біт, у той же час, кодуватиметься відсутністю такого символу.

Алгоритм методу доволі простий і зрозумілий. Під час вбудовування стеганоповідомлення порожній текстовий файл-контейнер зчитується рядками. При цьому у кінці кожного рядка вилучаються всі “пропускні” символи (пропуски, знаки табуляції, символи повернення каретки і нового рядка). Згодом, залежно від значення поточного біта стеганоповідомлення, поданого у вигляді двійкового вектора, приймається рішення про дописування чи не дописування в кінець рядка символу пропуску. Модифікований рядок записується у файл-результат.

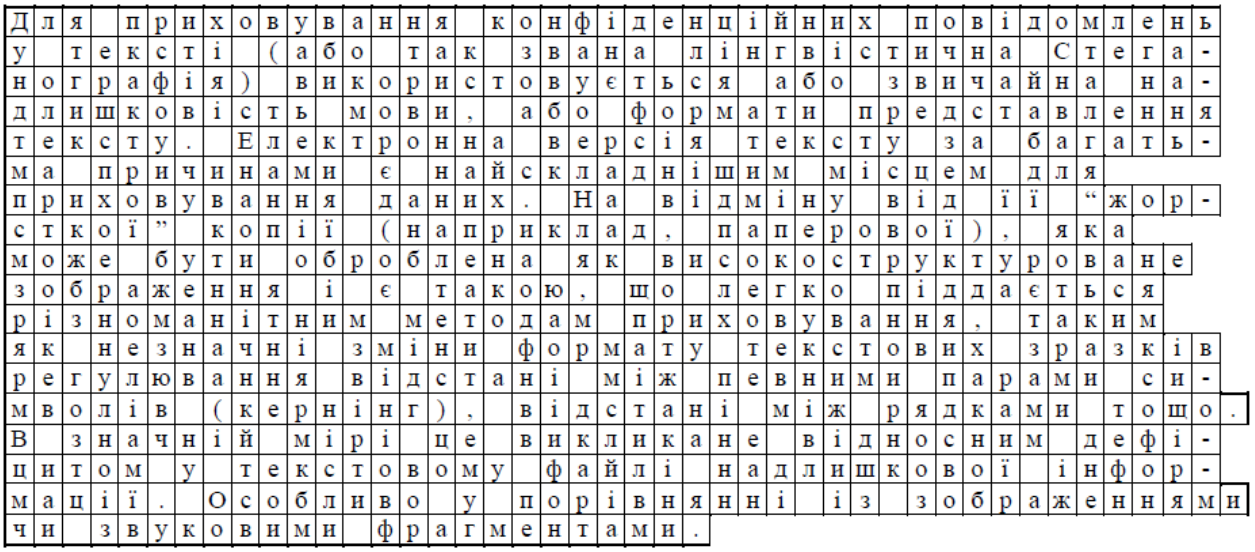



Рисунок 2.3 – Фрагмент текстового контейнера, використаного для приховування даних

На рис. 2.4-2.6 подано послідовність MathCAD-команд, за допомогою яких реалізовується алгоритм методу хвостових пропусків.

```
CC := READBIN("D:\M_TEX7.TXT" , "byte" )

M := "Алгоритм"

D2B(x) := | for i ∈ 1..8
          |     V_i ← mod(x,2)
          |     x ← floor(x/2)
          | V
```

Рисунок 2.4 – Програмний код – переведення ASCII-коду символа з десяткового формату у двійковий

```
C := | i ← 1
     | while i ≤ rows(CC)
     |     C_(rows(C)+1) ← CC_i  if CC_i ≠ 13
     |     if CC_i = 13
     |         k ← 0
     |         j ← i
     |         while CC_(j−1) = 32
     |             k ← k + 1
     |             j ← j − 1
     |         C ← submatrix(C, 1, rows(C) − k, 1, 1)
     |         C_(rows(C)+1) ← 13
     |     i ← i + 1
     | C
```

Рисунок 2.5 – Програмний код – вилучення “пропускних” символів у кінці кожного рядка порожнього контейнера

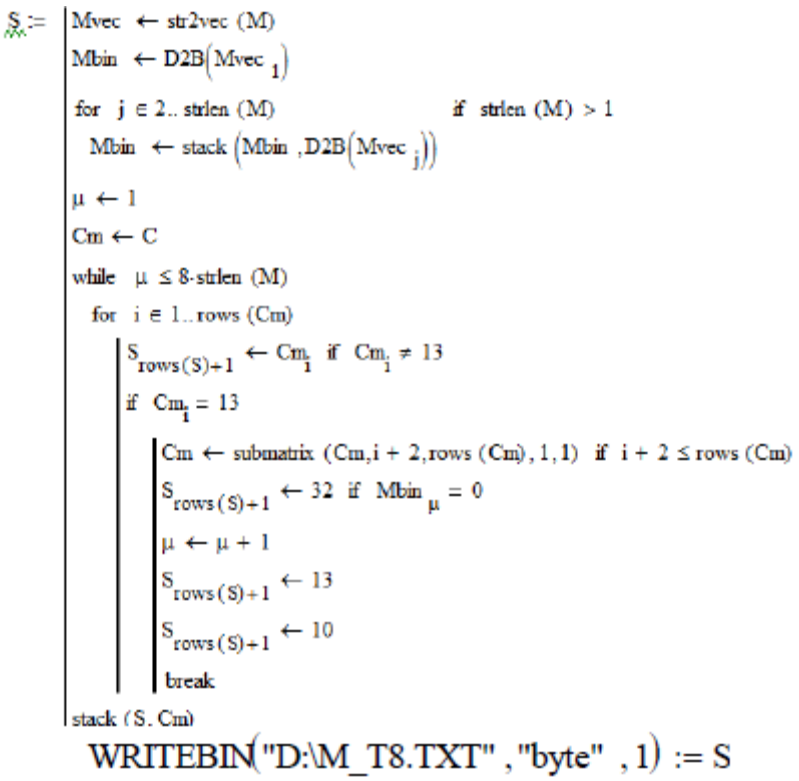



Рисунок 2.6 – Програмний код – вбудовування у контейнер даних методом хвостових пропусків

Під час видобування стеганоповідомлення із заповненого контейнера файл зчитується рядками, а значення поточного біта стеганоповідомлення встановлюється на основі наявності або відсутності символу пропуску в кінці рядка (рис. 2.7-2.8).

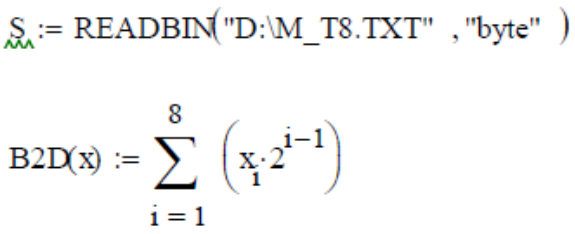



Рисунок 2.7 – Програмний код – переведення ASCII-коду з двійкового формату у десятковий

```
M := | μ ← 1
     | for i ∈ 1..rows(S)
     |   if S_i = 13
     |     k ← 0
     |     j ← i
     |     while S_(j−1) = 32
     |       k ← k + 1
     |       j ← j − 1
     |     Mbin_μ ← 1 if k = 0
     |     Mbin_μ ← 0 if k = 1
     |     μ ← μ + 1
     | for j ∈ 1..rows(Mbin) ÷ 8
     |   kod_sym ← B2D(submatrix(Mbin, 8·j − 7, 8·j, 1, 1))
     |   break if kod_sym = 0
     |   Mvec_j ← kod_sym
     | Mvec

WRITEBIN("D:\M_T88.TXT", "byte", 1) := M
```

Рисунок 2.8 – Програмний код – видобування методом хвостових пропусків конфіденційного повідомлення з текстового контейнера

На рис. 2.9 подано фрагмент заповненого текстового контейнера, у якому приховано два перші символи повідомлення ”Алгоритм”, а саме: символ ”А”, двійковий формат ASCII-коду якого D2B(str2vec("A"))$^T$ = ($0_{LSB}$ 0 0 0 0 0 1 $1_{MSB}$) та символ ”л” – D2B(str2vec("л"))$^T$ = ($1_{LSB}$ 1 0 1 0 1 1 $1_{MSB}$). Для пояснення у дохідливій формі суті алгоритму використано схожий до попереднього методу підхід: у кінці рядків зафарбованими клітинками позначено символи пропуску, що містять нульові біти двійкового вектора секретного повідомлення. Одиничні біти приховують в собі інші рядки, а саме ті, останні символи яких відрізняються від символу пропуск. Наприклад, сьомий-десятий.

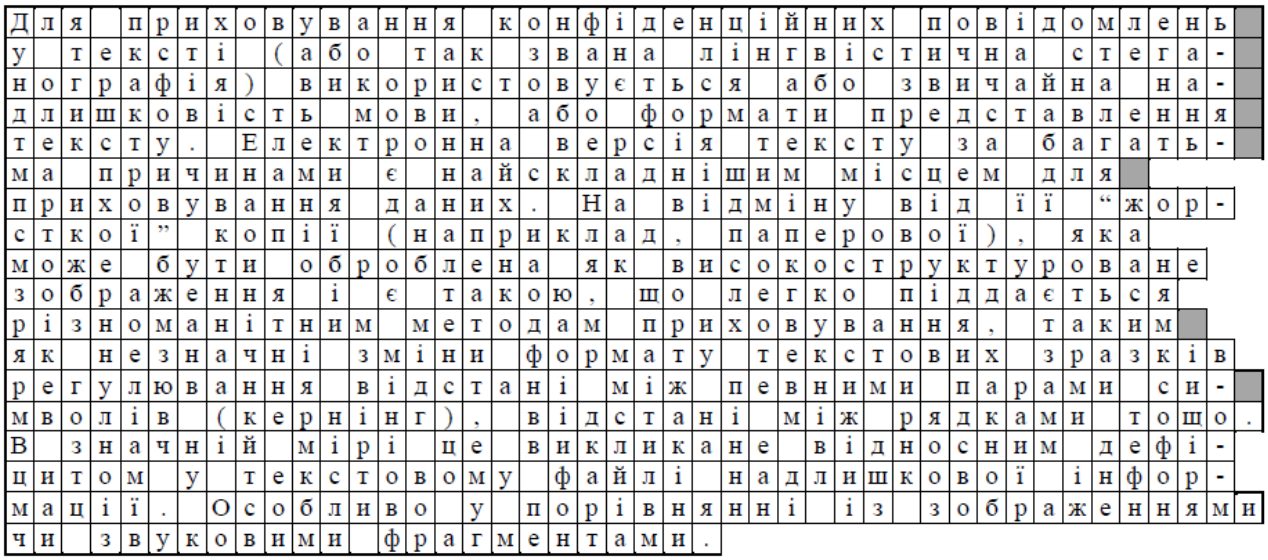



Рисунок 2.9 – Фрагмент заповненого методом хвостових пропусків текстового контейнера

Переваги методу хвостових пропусків: він може бути застосований до будь-якого тексту; зміни у форматі є досить непомітними, оскільки вільні місця, що використовуються, є периферійними по відношенню до основного тексту.

Пропускна здатність цього методу сильно залежить від використовуваного файлу-контейнера: чим більше у файлі рядків, тим більшу кількість інформації в ньому можна приховати. Якщо прийняти вищеописаний формат файлу-контейнера (кожен рядок має по 80 символів), пропускна здатність побудованої стеганосистеми становитиме 0,15 %. Слід зауважити, що роздрукування файлу-результату призводить до втрати прихованої інформації: на папері або іншому твердому носії неможливо виявити пропуски в кінці рядків. Крім того недоліком цього методу (як і попереднього) є те, що деякі текстові редактори самі по собі додають пропуски в кінець рядків. У той же час, цей метод можна визнати прийнятним для практичного застосування, особливо, якщо використати попереднє стиснення і шифрування стеганоповідомлення.

**Модифікований метод хвостових пропусків.** Цей метод полягає у дописуванні в кінець кожного рядка текстового контейнера від 0 до 15 символів пропуску, залежно від значення (у десятковому еквіваленті) півбайта ASCII-коду приховуваного символу, поданого у двійковому форматі. Таким чином, для приховування одного символу стеганоповідомлення досить двох рядків файлу-контейнера.


Рисунок 2.10 – Фрагмент оригіналу тексту контейнера, використаного для приховування повідомлення

Загалом, алгоритм приховування та вилучення стеганоповідомлення не суттєво відрізняється від методу зміни кількості пропусків у кінці текстових рядків. Варіант його реалізації у середовищі MathCAD подано на рис. 2.11-2.12.

$$CC := READBIN("D:\M\_TEX8.TXT" , "byte" )$$

$$M := "Алгоритм"$$

$$B2D(x) := \sum_{i=1}^{4} \left( x_i \cdot 2^{i-1} \right)$$

Рисунок 2.11 – Програмний код – переведення півбайта ASCII-коду символу з двійкового формату у десятковий

Далі іде програмний код вилучення “пропускних” символів у кінці кожного рядка порожнього контейнера. Він такий самий як у попередньому методі (рис. 2.5), як і процедура переведення ASCII-коду символа з десяткового формату у двійковий (рис. 2.4).

Видобути стеганоповідомлення з заповненого контейнера можна так, як показано на рис. 2.13.

```
S :=  Mvec ← str2vec(M)
      Mbin ← D2B(Mvec_1)
      for j ∈ 2..strlen(M)                    if strlen(M) > 1
        Mbin ← stack(Mbin, D2B(Mvec_j))
      μ ← 1
      Cm ← C
      while μ ≤ 2·strlen(M)
        for i ∈ 1..rows(Cm)
          S_(rows(S)+1) ← Cm_i  if Cm_i ≠ 13
          if Cm_i = 13
            Cm ← submatrix(Cm, i + 2, rows(Cm), 1, 1)  if i + 2 ≤ rows(Cm)
            ch ← B2D(submatrix(Mbin, 4·μ − 3, 4·μ, 1, 1))
            for j ∈ 1..ch          if ch ≠ 0
              S_(rows(S)+1) ← 32
            μ ← μ + 1
            S_(rows(S)+1) ← 13
            S_(rows(S)+1) ← 10
            break
      stack(S, Cm)

      WRITEBIN("D:\M_T9.TXT", "byte", 1) := S
```

Рисунок 2.12 – Програмний код – вбудовування у контейнер конфіденційного повідомлення модифікованим методом хвостових пропусків

```
S := READBIN("D:\M_T9.TXT" , "byte" )

B2D(x) := Σ_{i=1}^{8} ( x_i · 2^{i−1} )

D2B(x) := | for i ∈ 1..4
          |     V_i ← mod(x, 2)
          |     x ← floor(x / 2)
          | V

M := | μ ← 1
     | for i ∈ 1..rows(S)
     |   if S_i = 13
     |     k ← 0
     |     j ← i
     |     while S_{j−1} = 32
     |       k ← k + 1
     |       j ← j − 1
     |     Mbin1 ← D2B(k)
     |     Mbin ← Mbin1  if μ = 1
     |     Mbin ← stack(Mbin, Mbin1)  if μ > 1
     |     μ ← μ + 1
     | for j ∈ 1..rows(Mbin) ÷ 8
     |     kod_sym ← B2D(submatrix(Mbin, 8·j − 7, 8·j, 1, 1))
     |     break if kod_sym = 0
     |     Mvec_j ← kod_sym
     | Mvec

WRITEBIN("D:\M_T0.TXT" , "byte" , 1) := M
```

Рисунок 2.13 – Програмний код – видобування модифікованим методом хвостових пропусків конфіденційного повідомлення з текстового контейнера

На рис. 2.14 подано результат вбудовування у текстовий контейнер перших чотирьох символів повідомлення “Алгоритм”. Нагадаємо, двійкове подання ASCII-коду символу “А” таке: $0_{LSB}$ 0 0 0 0 0 1 $1_{MSB}$. Розіб'ємо його на два півбайти: $0_{LSB}$ 0 0 0 $_{MSB}$ та 0 $_{LSB}$ 0 1 $1_{MSB}$, десяткове значення яких 0 і 12 відповідно. Порівнюючи рис. 2.10 та рис. 2.14, бачимо, що у кінці другого рядка (рис. 2.14) з’явилося дванадцять пропусків, а в кінці першого рядка вони відсутні. Такий стан справи повністю узгоджується з попередніми міркуваннями. Щоб у текстовому контейнері (рис. 2.10) приховати символ ”А”, слід дописати нуль та дванадцять символів “пропуск” в кінці першого та другого рядка відповідно. Аналогічно,

D2B(str2vec("л"))$^{\mathbf{T}}$ = ($1_{LSB}$ 1 0 1 0 1 1 $1_{MSB}$) складається з двох півбайтів $1_{LSB}$ 1 0 1 $_{MSB}$ , 0 $_{LSB}$ 1 1 $1_{MSB}$, у десятковому форматі значення яких 11 та 14. У зв’язку з цим, саме таку кількість символів “пропуск” (одинадцять і чотирнадцять) маємо у третьому та четвертому рядку файла-результату (рис. 2.14). Їх функція – приховувати символ “л” повідомлення.

Переваги та недоліки модифікованого методу хвостових пропусків ті ж, що і в попередньому випадку. Пропускна здатність – 0,63 %. Збільшення у порівняні з методом хвостових пропусків пропускної здатності призводить до того, що стеганограму можна виявити легше.


Рисунок 2.14 – Фрагмент заповненого модифікованим методом хвостових пропусків текстового контейнера

У таблицю 2.1 зведемо відомості, що стосуються пропускної здатності розглянутих методів та можливості видобування прихованої інформації з паперових носіїв.

Таблиця 2.1 – Порівняльний аналіз методів щодо пропускної здатності та можливості видобування прихованої інформації з паперових носіїв

| № з/п | Назва методу | Пропускна здатність | Можливість видобування прихованої інформації з паперових носіїв |
|---|---|---|---|
| 1. | Метод зміни інтервалу між реченнями | 0,08% | + |
| 2. | Метод хвостових пропусків | 0,15% | – |
| 3. | Модифікований метод хвостових пропусків | 0,63% | – |

## 2.3 Контрольні запитання та завдання:

1. Назвіть методи текстової стеганографії.
2. Охарактеризуйте метод зміни порядку проходження маркерів кінця рядка CR/LF.
3. У чому полягає метод форматування тексту за допомогою пробілів?
4. Охарактеризуйте метод хвостових пробілів.
5. Охарактеризуйте метод двійкових нулів.
6. Охарактеризуйте метод чергування маркерів кінця.
7. Охарактеризуйте метод знаків однакового накреслення.
8. Назвіть основні порівняння методів текстової стеганографії.

**Завдання.**

Вивчити теоретичний матеріал щодо функціонування та створення стеганографічних систем приховання повідомлень в текстових файлах, ознайомитись з математичними моделями опису даних методів.

Скласти програми реалізації приховання повідомлень в текстових файлах та провести дослідження їх продуктивності та стеганостійкості.

Провести порівняльний аналіз методів, і навести приклади їх практичного використання.

Електронний навчальний документ

# МЕТОДИЧНІ ВКАЗІВКИ
до практичних занять

з дисципліни
# СТЕГАНОГРАФІЧНІ СИСТЕМИ

для студентів усіх форм навчання
спеціальності 125 Кібербезпека
Освітня програма
"Безпека інформаційних та комунікаційних систем”

Упорядник Федюшин Олександр Іванович

Відповідальний випусковий Г.З. Халімов

Авторська редакція