Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет __*Інфокомунікацій*_____
(повна назва)
Кафедра _____*Інформаційно-мережної інженерії*_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

Методи захисту графічного та відеоконтенту у веб-середовищі від несанкціонованого використання
(тема)

Виконав:
студент _2_ курсу, групи __*ІМІм-21-1*___
Іноземцев Є.В.
(прізвище, ініціали)

Спеціальність _*172 Телекомунікації та радіотехніка*_____
(код і повна назва спеціальності)
Тип програми ____*освітньо-професійна*___
(освітньо-професійна або освітньо-наукова)
Освітня програма _*Інформаційно-мережна інженерія*_____
(повна назва освітньої програми)

Керівник ____*ст.викл. Твердохліб В.В.*_____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ *Безрук В.М.*
(підпис) (прізвище, ініціали)
2022 р.

Не містить відомостей, заборонених
до відкритого публікування

Керівник_____ /*В.В.Твердохліб*

Студент _____ / *Є.В.Іноземцев*

Харківський національний університет радіоелектроніки

Факультет *Інфокомунікацій*

Кафедра *Інформаційно-мережної інженерії*

Рівень вищої освіти другий (магістерський)

Спеціальність *172. Телекомунікації та радіотехніка*

(код і повна назва)

Тип програми *Освітньо-професійна*

(освітньо-професійна або освітньо-наукова)

Освітня програма *Інформаційно-мережна інженерія*

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_21_» *жовтня* 20 22 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові *Іноземцеву Єгору Владиславовичу*

(прізвище, ім'я, по батькові)

1. Тема роботи *Методи захисту графічного та відеоконтенту у веб-середовищі від несанкціонованого використання*

затверджена наказом університету від _21_ __жовтня__2022 р. № __1376 Ст__

2. Термін подання студентом роботи до екзаменаційної комісії __ _грудня_ 2022 р.

3. Вихідні дані до роботи *Обгрунтувати необхідність захисту графічного та відеоконтенту на базі застосування цифрових міток. Дослідити базові підходи до застосування стеганографічних методів для внесення цифрових міток. Виконанати дослідження традиційних підходів до розміщення цифрових міток у межах jpeg-контейнерів. Дослідити підхід, у рамках якого реалізація LSB-методу є контентно-орієнтованою. Розробити концепцію підходу, що орієнтований на захист відео контенту на базі внесення цифрових міток в окремі кадри потоку.*

4. Перелік питань, що потрібно опрацювати в роботі *Вступ*

1 . Загальні питання захисту інформаційних продуктів та електронних документів

2. Існуючі механізми захисту інформаційних продуктів у базисі застосування стеганографічних алгоритмів.

3. Побудова механізму внесення цвз для статичних об'єктів без фронтального заповнення

4. Розробка підходів до внесення цвз на випадок маркування відеоконтенту

Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
слайди презентації в форматі Power Point (назва та мета роботи, сутність методів захисту інформаційних продуктів та електронних документів на базі ЦВЗ, поширені механізми захисту інформаційних продуктів на базі стеганографічних алгоритмів; механізм внесення ЦВЗ для статичних об'єктів без фронтального заповнення, внесенні ЦВЗ у відео, висновки)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вступ		
2	Загальні питання захисту інформаційних продуктів та електронних документів		
3	Існуючі механізми захисту інформаційних продуктів у базисі застосування стеганографічних алгоритмів		
4	Побудова механізму внесення цвз для статичних об'єктів без фронтального заповнення		
5	Розробка підходів до внесення цвз на випадок маркування відеоконтенту		
6	Висновки		
7	Оформлення пояснювальної записки		

Дата видачі завдання *21 жовтня* 2022 р.

Студент _____
(підпис)

Керівник роботи _____ *ст. викл. Твердохліб В.В.*
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 61 с., 15 рис., 24 джерела, 1 додаток

ЦИФРОВИЙ ВОДЯНИЙ ЗНАК, JPEG, LSB, коефіцієнт ДКП, H.264, MPEG

Об'єкт дослідження – методи захисту мультимедійного контенту від несанкціонованого використання на базі стеганографічних алгоритмів.

Мета роботи – дослідження можливостей удосконалення існуючих методів захисту мультимедійного контенту на базі існуючих стеганографічних алгоритмів за рахунок їх доробки.

Виконується огляд методів захисту даних на основі використання цифрових міток. Досліджуються поширені методи внесення цифрових водяних знаків у графічні та відеодані Виявляються недоліки стандартизованих підходів. Виконується дослідження нефронтального способу заповнення контейнерів. Розробляються підходи щодо удосконалення алгоритмів, орієнтованих на інкапсуляцію даних у простір найменш значущих біт для випадків графічного та відео контенту.

THE ABSTRACT

Explanatory note: 61p., 15 fig., 25 sources, 1 app.

DIGITAL WATERMARK, JPEG, LSB, DCP COEFFICIENT, H.264, MPEG

The object of research is methods of protecting multimedia content from unauthorized use based on steganographic algorithms.

The purpose of the work is to investigate the possibilities of improving existing methods of protecting multimedia content based on existing steganographic algorithms by improving them.

A review of data protection methods based on the use of digital labels is performed. Common methods of applying digital watermarks to graphic and video data are investigated. Deficiencies of standardized approaches are revealed. A non-frontal way of filling containers is being researched. Approaches are being developed to improve algorithms focused on data encapsulation in the space of least significant bits for cases of graphic and video content..

ЗМІСТ

С.

ПЕРЕЛІК СКОРОЧЕНЬ

LSB– (Least Significant Bit) – найменш значимий біт;
НЗБ –найменш значимий біт;
ЦВЗ – цифровий водяний знак;
ЕО – електронний об’єкт;
MPEG – сімейство стандартів відео кодування;
BMP – формат представлення некодованих графічних даних;
JPEG – технологія кодування зображень;
ПЗ – програмне забезпечення;
HASP – (Hardware Against Software Piracy) – апаратно-програмний модуль захисту програмного забезпечення;
ДЗЗ – дистанційне зондування Землі;
YUV – формат яскравісно-хроматичного опису зображення;
YCbCr – формат яскравісно-хроматичного опису зображення;
ДКП – дискретно-косинусне перетворення;
H.265/HEVC– (high-efficiency video coding) — технологія високоефективного кодування відео H.265;
H.265/ AVC – (advavced video coding) — технологія високоефективного кодування відео H.264;

ВСТУП

В умовах постійного стрімкого зростання цифрового контенту, у загальному його обсязі збільшується частка даних, що потребують як захисту від неавторизованого використання, підробок чи клонування, так і застосування механізмів стеження за їх розповсюдженням.

Найбільшою мірою це стосується мультимедійних продуктів до яких, з одного боку, у наслідок їх специфіки не може бути застосовано більшість механізмів захисту, що є ефективними, наприклад, для програмного забезпечення [1, 2].

З іншого боку, роль мультимедіа, як галузі, що початково створювалася як елемент індустрії розваг, стає все більш значущою. Сьогодні вона стосується таких галузей, як:

- освіта та навчання;
- соціальні комунікації;
- засоби масової інформації;
- ведення бізнесу тощо.

Тобто, все це загострює проблематику, пов’язану з необхідністю забезпечення захищеності даних цього типу.

Ще одним аспектом, який сьогодні актуалізує усі питання захисту мультимедіа, є обмеженість та низька ефективність поширених традиційних підходів. Зокрема, це зумовлено тим, що:

1. Ряд традиційних підходів до цифрового захисту мультимедійних продуктів орієнтується на той чи інших тип файлу (swf, gif, pdf, avi, mp4 тощо), використовуючи локально реалізовані механізми.

Продуктивність таких механізмів на сьогодні, в умовах, коли потенційний зловмисник має у розпорядженні суттєві обчислювальні можливості та доступ до повного опису структури файлів, гарантуватися не може категорично.

2. Перші релізи уніфікованих методів захисту, такі, наприклад, як використання водяних знаків у видимій області цифрового документу, можуть нівелюватися зловмисником шляхом реконструкції маркованих зон.

Водночас, такі методи виявили свою перспективність. Їхня ключова відмінність – орієнтування не на певний формат файлу та його особливості, а на початковий насій даних, яким на випадок мультимедіа є графічні або аудіо об’єкти.

У той же час, використання стеганографічних підходів для внесення цифрових міток дозволило суттєво збільшити захищеність даних. При цьому, єдиними, але суттєвими недоліками таких підходів є, по-перше, відсутність на сьогодні єдиних стеганографічних стандартів, а по-друге – недостатня ефективність ряду поширених методів. За таких умов усі питання, що стосуються дослідження, розробки та впровадження методів та алгоритмів захисту даних на базі стеганографічного піходу є актуальними.

1 ЗАГАЛЬНІ ПИТАННЯ ЗАХИСТУ ІНФОРМАЦІЙНИХ ПРОДУКТІВ ТА ЕЛЕКТРОННИХ ДОКУМЕНТІВ

1.1 Базові класи інформаційних продуктів

У загальному випадку інформаційним продуктом (інфопродуктом) сьогодні вважається будь-який контент, створений з використанням інформаційних технологій та інструментів, побудованих на їх базі.

Серед найбільш поширених сьогодні інфопродуктів зараз можна виділити такі, як [1-3]:

- програмні продукти (додатки) та їх компоненти для локального застосування на рівні кінцевого пристрою, або мережеві;
- продукти, що існують у вигляді мультимедійних конструкцій, відео або графічному;
- продукти текстової природи - самостійні або такі, що є супутніми іншим інфопродуктам. Нерідко для забезпечення первинного захисту розповсюджуються у вигляді файлів djvu чи pdf.

По мірі збільшення частки інформаційних продуктів, а в першу чергу – після того, як головним каналом їх розповсюдження стала Всесвітня мережа – питання їх захисту перетворилося на одне з ключових завдань, що потребують негайного вирішення. Це зумовлено, у першу чергу, тим, що більшість інфопродуктів є об’єктами авторського права та комерційної таємниці.

Водночас, в існуючих умовах весь перелік ймовірних переваг, що забезпечується представленням та розповсюдженням інфопродуктів мережевим середовищем, зводиться нанівець легкістю, з якою їх може бути викрадено або модифіковано.

Відповідно, для ПЗ необхідно забезпечити захист від несанкціонованого використання, відтворення та модифікації, тоді як для інфопродуктів інших типів серед зазначених першочергово важливим є гарантування їх автентичності.

При цьому, найвищий рівень захисту від копіювання, модифікації та неавторизованого використання мають програмні продукти, як наслідок того, що [4]:

- комерційне ПЗ не розповсюджується з відкритим кодом, що, у свою чергу, перетворює процес його модифікації та/або відтворення на надмірно

складний а у деяких випадках – на цілком недоцільний процес з фінансової точки зору;

- захист програмного забезпечення від копіювання та несанкціонованого застосування передбачає використання програмних та/або апаратних (HASP) ключів;

- виконується прив’язка копії комерційного програмного засобу до цифрового образу системи користувача за ідеологією Finger Printing, відтак використання цієї ж копії на іншому пристрої стає практично неможливим.

1.2 Електронні документи

У широкому розумінні поняття «електронний документ» може розумітися як [1, 3]:

1. Той чи інший документ, який відпочатку являє собою фізичний об’єкт та у наслідок проведення процедур оцифрування отримує електронний вигляд. Прикладами таких документів є:

- накладна чи платіжне доручення, які містять підписи уповноважених осіб та печатки;

- скани паспорту та інших документів особи;

- свідотства на право власності тощо.

Зрозуміло, що використання таких документів замість оригіналів має суттєві обмеження, позаяк має гарантуватися їх автентичність.

Тобто, потрібно забезпечити захист таких документів від можливої модифікації зловмисником.

2. Документ, що початково має електронну природу. Прикладами таких документів є:

- цифрові квитанції, що застосовуються у ході операцій з крипто валютами;

- електронні транзакції систем інтернет-банкінгу;

- біометрія, що застосовується для аутентифікації особи.

У загальному випадку дані класи документів можуть вважатися захищеними, а відтак додаткових заходів з безпеки не потребують.

1.3 Ключові підходи до захисту інфопродуктів та електронних документів

Значна частина поширених сьогодні методів захисту для забезпечення автентичності електронних об’єктів (ЕО) – як електронних документів, так і інформаційних продуктів, базується на використанні стеганографічних алгоритмів і реалізується за такими напрямками, як [1, 3-5]:

- watermarking, або внесення цифрових водяних знаків (ЦВЗ), для забезпечення захисту авторського права та/або права власності на той чи інший ЕО;

- finger printing, або вбудовування унікальних ідентифікаційних номерів;

- captioning, або внесення заголовків.

У свою чергу, серед існуючого переліку реалізацій зазначених напрямків стеганографічного захисту ЕО найбільш ефективним є такі, що базуються на розміщенні усередину ЕО, що підлягає захисту, зовні непомітних цифрових міток, або цифрових водяних знаків - ЦВЗ.

При цьому, зчитування та аналіз ЦВЗ і далі – прийняття рішення про автентичність того чи іншого ЕО виконується спеціалізованим декодером. ЦВЗ для таких випадків може містити деякий код, інформацію щодо власника, мітку про автентичність документу від уповноважених органів тощо.

Водночас, внесення ідентифікаційної інформації виробником інфопродуктів, у сутності, нічим на відрізняється від внесення ЦВЗ. Єдина відмінність полягає у використанні для кожної копії унікального ідентифікатору. Це, у свою чергу, дає можливість виробникові відстежувати рух свого продукту, зокрема, відстежувати випадки незаконного тиражування тощо.

Стосовно напрямку captioning, або, інакше кажучи, внесення заголовків, слід зазначити, що його ключове завдання є дещо відміннім від попередньо розглянутих випадків, а саме - забезпечити зберігання даних різного типу у вигляді єдиного цілого.

Наприклад, заголовки можуть вноситися для підпису медичних знімків, нанесення умовних позначень на схеми, карти тощо.

1.4 Принципи реалізації методів внесення цифрових міток для ЕО

З точки зору стеганографії, як окремої специфічної галузі, сам ЕО, який підлягає захисту внесенням цифрової мітки, *контейнером*, а сам ЦВЗ - корисним навантаженням.

Сам процес розміщення та зчитування міток може бути проілюстровано схемою на рис. 1.1 [4, 6].

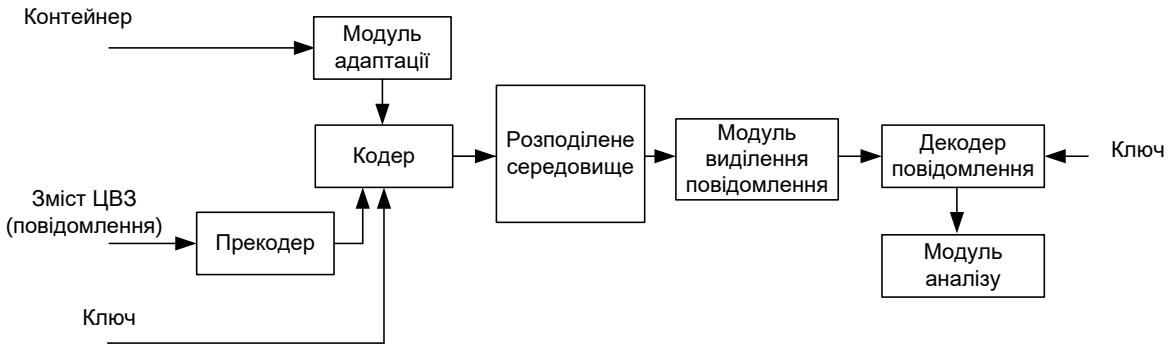


Рисунок 1.1 – Загальний сценарій розміщення та зчитування міток ЦВЗ

Так, на першому етапі контейнер (ЕО чи його окрема частина) приводяться до вигляду, який безпосередньо дозволяє виконувати внесення мітки. На схемі 1.1 за це відповідає модуль адаптації.

Далі інформація, яку має нести ЦВЗ, конвертується до формату, який може бути інкапсульовано (вбудовано) у контейнер. Цю процедуру реалізує прекодер. Тут передбачається щонайменше такі режими реалізації даного технологічного етапу, як:

- виключно конвертація даних ЦВЗ до певного формату;
- конвертація з застосуванням шифрування.

Після процедур прекодінгу та попередньої адаптації контейнеру (ЕО) на базі кодеру виконується процедура інкапсулювання, у ході якої *стеганографічний ключ*, або *стегоключ*, задає параметри та особливості реалізації процедури.

Електронний об’єкт перебуваючи у загальному розділеному середовищі, може бути у довільний спосіб тиражовано та/або неодноразово переміщено чи модифіковано. Для перевірки його цілісності та автентичності на боці вендору може бути ініційовано процедуру перевірки, у ході чого вбудоване повідомлення (ЦВЗ) з електронного об’єкту зчитується та конвертується до вигляду, який надалі може бути піддано аналізу.

За результатами аналізу ЦВЗ робиться висновок про цілісність та автентичність ЕО.

При цьому зазначимо, що зловмисник має технічну можливість виявлення, зчитування та розшифрування цифрової мітки. За таких умов він має змогу:

- вилучення ЦВЗ зі складу ЕО, що формально нівелює права вендора на даний інфопродукт;

- модифікації та тиражування цифрових міток, що на випадок цифрових документів веде до їх компрометації а для інформаційних продуктів – фактично сприяє «узаконенню» їх крадіжки;

- компрометації вендора, що може бути реалізовано шляхом розміщення його ЦВЗ у складі сторонніх ЕО, до яких він не має жодного відношення.

Звідси можемо сформулювати загальні вимоги щодо принципів вбудовування цифрових міток [1, 5, 6]:

- необхідно мінімізувати ймовірність виявлення самого факту внесення ЦВЗ;

- складність процесу вилучення цифрової мітки за умови її виявлення має бути такою, що зробить його недоцільним;

- на базі виявленого та розшифрованого фрагменту мітки зловмисник не повинен мати змоги для її повної реконструкції.

Дані вимоги є додатковими до загальний вимог щодо побудови стеганографічних систем.

1.5 Вимоги до побудови стеганографічних систем

Серед багатьох існуючих вимог сьогодні найбільш суттєвими є 3 головні, дотримання яких у загальному випадку здатне гарантувати ефективність стегосистеми у цілому, не зважаючи на ті чи інші особливості її реалізації. Такі вимоги стосуються:

- продуктивність та раціональність побудови стегосистеми;

- ступінь захищеності від виявлення, що стосується як ЦВЗ, так і будь-яких прихованих даних іншого характеру у цілому;

- фактична ємність стегосистеми.

У даному випадку ємність V стегосистеми розглядається як відношення фактичного об’єму біт V_m інкапсульованих даних (повідомлення) до обсягу біт V_c самого контейнеру, за якого величина ступеню P захищеності буде на рівні не меншому, ніж деякий необхідний рівень P', а саме:

$$V = \frac{V_m}{V_c} \times 100\% \mid P \geq P', \tag{1.1}$$

Водночас, сутність терміну ступеня P захисту у залежності від ймовірності виявлення має різні трактування.

Так, відповідно до вимог щодо стегосистеми з огляду на галузь її застосування поняття ступеню захисту може розумітися як [2, 5, 6]:

- величина ймовірності виявлення ЦВЗ чи будь-яких інших типів вбудовувань зловмисником;

- ймовірний час, який зловмисник витратить на виявлення ознак заповненості контейнеру.

Разом з тим, так як для стегосистеми у загальному випадку першочергово важливим є збереження таємниці самого факту присутності вбудованих даних, при розгляді показника P не ураховується ймовірність (або час) виокремлення, розшифрування та реконструювання змісту прихованих даних.

У свою чергу, параметр продуктивності G стеганографічної системи відображає її можливість щодо обробки тієї чи іншої кількості V_m біт повідомлення, яке вбудовується у контейнер, протягом одиниці часу t, що еквівалентно виразу:

$$G \equiv \frac{V_m}{t}. \tag{1.2}$$

Окрім цього, існує взаємозв’язок між параметрами P захищеності та можливим наповненням V контейнеру, а саме:

$$V \uparrow \rightarrow P \downarrow. \tag{1.3}$$

Властивість (1.3) є однією з найбільш важливих для будь-якої стегосистеми. Це пов’язано з тим, що більшість алгоритмів виявлення вбудованих даних за виконання умови:

$$\frac{V_c}{V_m} \rightarrow 0 \tag{1.4}$$

не є ефективними.

Далі з урахуванням вимог до стегосистем виконаємо огляд найбільш поширених методів стеганографії та попередньо визначимо, які з них може бути використано для захисту електронних об’єктів.

2. ІСНУЮЧІ МЕХАНІЗМИ ЗАХИСТУ ІНФОРМАЦІЙНИХ ПРОДУКТІВ У БАЗИСІ ЗАСТОСУВАННЯ СТЕГАНОГРАФІЧНИХ АЛГОРИТМІВ

2.1 Обгрунтування відсутності єдиної ідеології захисту інфопродуктів на базі стеганографічних алгоритмів

Хоча на сьогодні більшість механізмів захисту електронних об’єктів базується на використанні стеганографічних алгоритмів, єдина парадигма стосовно їх застосування фактично відсутня. Це є наслідком того, що [4-6]:

- тип даних (зображення, аудіо чи відеопослідовності), що потребують захисту, визначає прийнятний діапазон алгоритмів, що можуть використовуватися;
- на сьогодні фактично відсутні єдині стандарти стеганографії;
- специфіка даних навіть у межах одного типу може суттєво впливати на можливість застосовуваності тих чи інших стеганографічних підходів (рис.2.1).

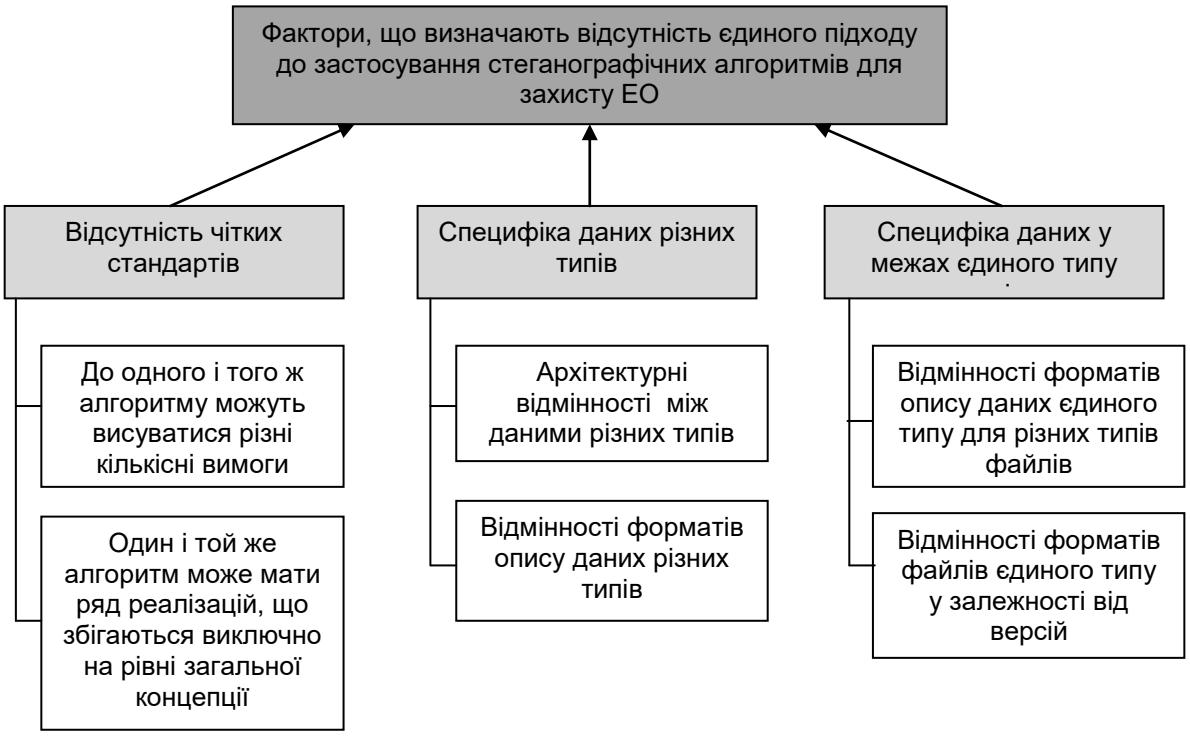


Рисунок 2.1 – Чинники, які визначають відсутність єдиної парадигми щодо застосування стеганографічних алгоритмів

Так, наприклад, механізми застосування одного і того ж алгоритму для випадку контейнерів хоча і одного типу, але утворених файлами різних форматів, різнитимуться.

Разом з тим, на відміну від випадків інкапсуляції прихованих повідомлень з подальшим надсиланням їх приймачеві, тобто, повноцінної стегосистеми, для захисту електронних об'єктів відсутні ряд характерних обмежень, що є притаманними стеганографічним системам у широкому розумінні, а саме:

- залежність (1.4) не є жорсткою; це зумовлено тим фактом, що кількість біт V_m, що є необхідними для внесення цифрових міток у стегоконтейнер, у загальному випадку буде апріорі суттєво нижчою, ніж для звичайної стегосистеми. Тобто, величина відношення величин V_c та V_m може сягати деякого невеликого значення V, що може бути проілюстровано виразом:

$$\left(\frac{V_c}{V_m}\right)_{\text{ЦВЗ}} \le V \le 0{,}6, \qquad (2.1)$$

у виразі (2.1) коефіцієнт 0,6 відповідає випадку наповненості контейнеру не більш, ніж на 60%. За таких умов більшість універсальних алгоритмів виявлення маскованих даних є неефективними [7];

- вимоги до параметру G продуктивності на випадок захисту ЕО є менш жорсткими, ніж для класичної стегосистеми.

Отже, у підсумку можна зазначити, що в умовах відсутності єдиного підходу до захисту інфопродуктів першочергово визначним для вибору того чи іншого стеганографічного алгоритму та особливостей його застосування є такі фактори, як:

- тип даних, що потребують захисту, та формат їх опису;

- архітектура алгоритму.

Відмінності в специфіці застосування алгоритмів стеганографічного захисту розглянемо на прикладі контейнерів bmp та jpeg форматів.

2.2 Підходи до захисту ЕО форматів bmp та jpeg

2.2.1 Обгрунтування необхідності стеганографічного захисту продуктів формату bmp та jpeg на базі ЦВЗ

Для системи прихованого зв'язку на базі стеганоалгоритмів, використання контейнерів bmp-типу є моветоном, що пов'язано з досить низько часткою файлів даного типу у мережевому трафіку у наслідок їх значного обсягу.

Таким чином, системи стегоаналізу апріорі сприймають bmp як потенційно підозрілі.

У нашому ж випадку файли bmp – це частковий випадок ЕО, які є об'єктом захисту з використанням ЦВЗ. Це, зокрема, графічні дані, які потребують найвищого рівня якості за будь-якими об'єктивними метриками а сааме [1, 6, 7]:

- професійні фотоматеріали з високою роздільною здатністю та високим рівнем глибини колірності, отримані з raw-файлів після видалення технічних опцій проведення зйомки;
- матеріали ДЗЗ;
- знімки, отримані з використанням медичних систем;
- матеріали, що є елементом біометрії.

Якщо ймовірність розповсюдження перелічених вище даних, поданих у bmp-форматі, у загальнодоступних файлосховищах є досить низькою, то розповсюдженість jpeg є на порядки вищою.

У свою чергу, на базі файлів jpeg можуть існувати як документи, що були попередньо переведені до цифрового вигляду, так і будь-які інші дані, що потребують захисту, але не мають жорстких вимог щодо збереження перинної якості, як на випадок bmp:

- електронні сертифікати;
- електронні книги;
- матеріали авторських навчальних курсів тощо.

2.2.2 Специфіка застосування одного і того ж механізму внесення ЦВЗ для файлів bmp та jpeg типів

Файл формату bmp, призначений для зберігання графічної інформації у некодованому вигляді, являє собою матрицю M на N пікселів $p_{k,\ell}$ [8].

Кожен з пікселів $p_{k,\ell}$, при цьому, за замовчуванням описується у 24-розрядному просторі RGB, тобто, є поєднанням трьох колірних каналів – відповідно, R, G та B, для кожного з яких резервується 8 розрядів (рис.2.2).

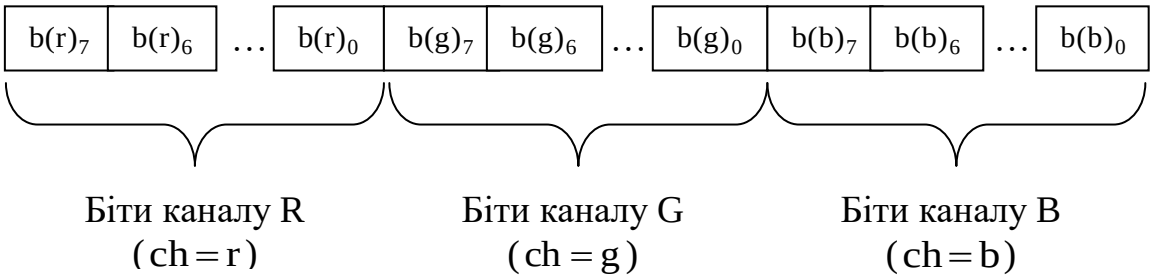


Рисунок 2.2 – Формат опису пікселя у межах bmp-файлу

При цьому, початковий опис будь-якого пікселя для bmp залишається незмінним. Таким чином, інкапсуляція даних ЦВЗ у контейнер даного типу може здійснюватися напряму, за законом конкретного використовуваного стегоалгоритму, та не потребує додаткових перетворень даних, а відміну від jpeg.

У свою чергу, jpeg є кодованим форматом, що утворюється на базі bmp шляхом виконання відносно нього ряду технологічних перетворень у наступній послідовності [9-11]:

1. Зміна колірного простору з BMP на якскавісно-хроматичний - YUV чи YCbCr.

2. Сегментація зображення з виокремленням блоків $\beta_{x,y}$ розмірністю 8x8 кожен та зміна формату колірного опису, при якому 4 компонентам Y ставиться у відповідність $n = \overline{1;4}$ хроматичних компонент.

3. Виконання процедури ортогонального перетворення блоків $\beta_{x,y}$ на базі дискретного косинусного перетворення (ДКП) окремо для блоків $\beta(ch)_{x,y}$ (ch=Y, Cb, Cr) тобто, блоків яскравості, та колірно-різницевих, або хроматичних - $\beta(Cb)_{x,y}$ і $\beta(Cr)_{x,y}$. У наслідок цього отримуємо трансформовані блоки $\hat{\beta}(ch)_{x,y}$, які містять спектральне представлення даних вихідних блоків $\beta(ch)_{x,y}$.

4. Квантування та округлення компонент $\eta(ch)_{\mu,\upsilon}$ блоку $\hat{\beta}(ch)_{x,y}$. Тут квантування, що являє собою процедуру поелементного ділення величин $\eta(ch)_{\mu,\upsilon}$ компонент на коефіцієнти $\gamma(ch)_{\mu,\upsilon}$ матриці квантування, реалізується за наступним принципом:

$$\breve{\eta}(ch)_{\mu,\upsilon} := \frac{\eta(ch)_{\mu,\upsilon}}{\gamma(ch)_{\mu,\upsilon}}; \quad \mu \vee \upsilon \uparrow \to \gamma \uparrow, \tag{2.2}$$

де $\breve{\eta}(ch)_{\mu,\upsilon}$ - квантова на компонента $\eta(ch)_{\mu,\upsilon}$

$\mu, \upsilon \in [0; 7]$ - координати компоненти $\eta(ch)_{\mu,\upsilon}$ блоку $\hat{\beta}(ch)_{x,y}$.

При цьому, найменшою мірою квантуються низькочастотні компоненти, яким відповідають мінімальні значення координат μ та υ. Відповідно, з ростом цих величин буде збільшуватися значення відповідних коефіцієнтів $\gamma(ch)_{\mu,\upsilon}$ квантування.

У свою чергу, процедура округлення будується на базі принципу, як показано наступним виразом:

$$\begin{cases} \breve{\eta}(ch)_{\mu,\upsilon} := 0 \,|\, \breve{\eta}(ch)_{\mu,\upsilon} < \zeta; \\ \breve{\eta}(ch)_{\mu,\upsilon} := \breve{\eta}(ch)_{\mu,\upsilon} \,|\, \breve{\eta}(ch)_{\mu,\upsilon} \geq \zeta, \end{cases} \tag{2.3}$$

де ζ - поріг квантування.

5. Лінеаризація компонент та кодування без втрат.

У ході даних процедур матриця 8x8 перетворюється на вектор з 674 компонент, які далі підлягають операціям ентропійного кодування.

Таким чином, для контейнеру jpeg прийнятним є лише один варіант застосування алгоритмів внесення ЦВЗ – а саме, після виконання усіх процедур конвеєру jpeg, які ведуть до незворотної втрати даних.

Отже, алгоритми внесення ЦВЗ можуть виконуватися виключно після процедур квантування та округлення (рис.2.3).

Далі проаналізуємо існуючі підходи до реалізації інкапсулювання даних (ЦВЗ та повідомлень у широкому розумінні) у контейнери

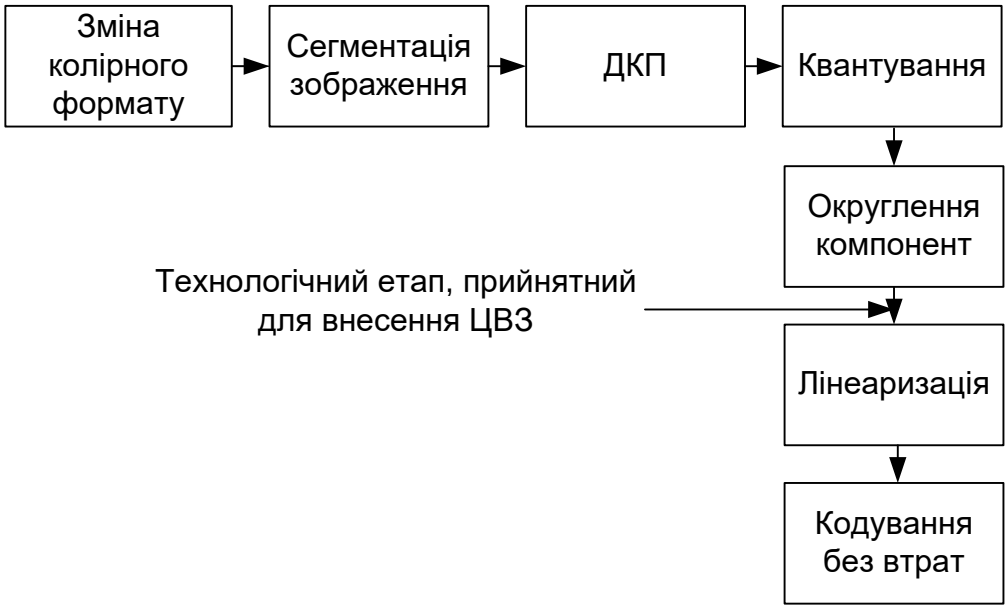


Рисунок 2.3 – Локалізація етапу внесення ЦВЗ на у загальній послідовності етапів JPEG-перетворення

2.3 Аналіз поширених механізмів внесення стеганографічних вбудовувань у контейнери

2.3.1 Інкапсуляції даних за умови використання принципу фронтального вбудовування

Фронтальний спосіб вбудовування біт секретного повідомлення має найпростішу реалізацію та вимагає найменший обсяг обчислювальної потужності апаратних пристроїв учасників інформаційної взаємодії.

Окрім цього, такому способу стеганографічного вбудовування характерними є такі властивості [4, 8, 12]:

- найвищий потенційно можливий рівень заповнення контейнеру;
- висока швидкість виконання процедури інкапсуляції та виокремлення вбудованих біт на прийомному боці.

Розглянемо принцип реалізації фронтального способу вбудовування за умов, що модифікація виконується на рівні біт.

При цьому, у ході вбудовування двійкових елементів повідомлення здійснюється зміна біт контейнеру Λ за принципом, наведеним виразом (2.4) та рис. 2.4.

$$\begin{cases} b_i := \neg b_{k\ell} \mid b_i \neq d_\omega; \\ b_i := b_{k\ell} \mid b_i = d_\omega, \end{cases} \tag{2.4}$$

де b_i - біт контейнеру, який задіюється у ході інкапсуляції;

d_ω - біт повідомлення, яких вбудовується.

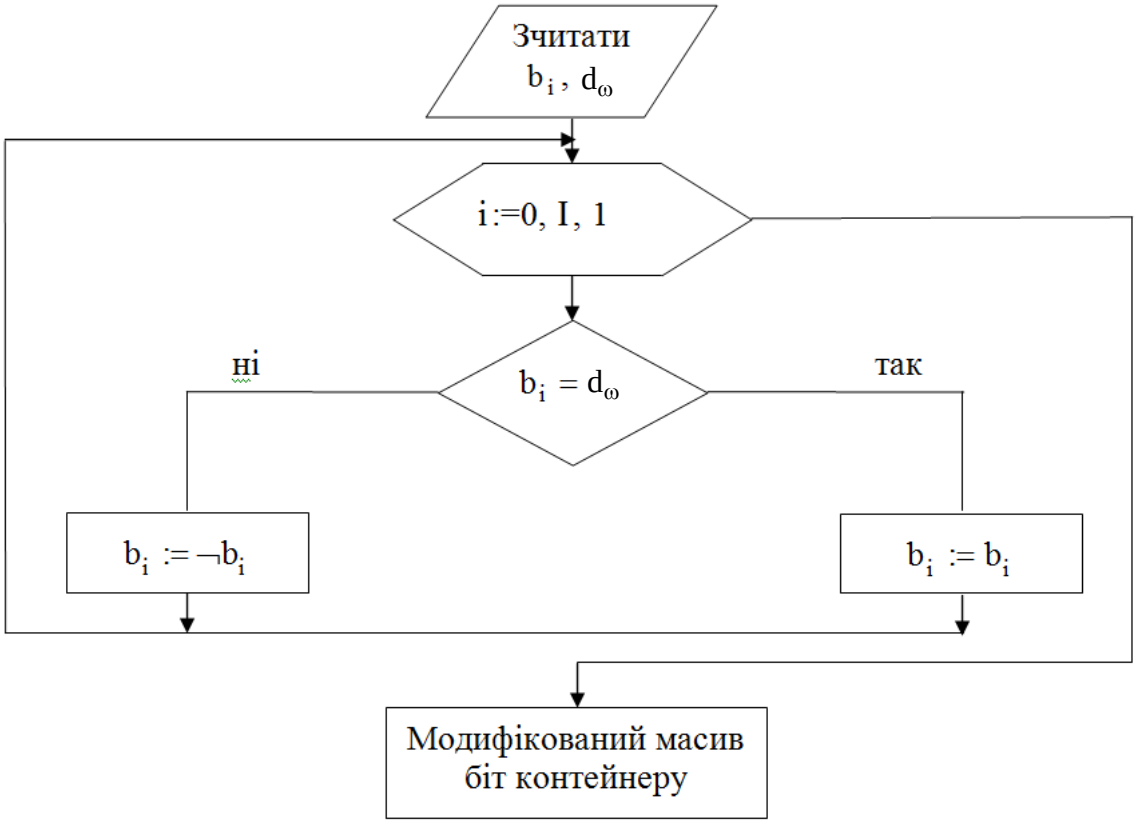


Рисунок 2.4 – Схематичне зображення правила зміни біт контейнеру під час інкапсуляції повідомлення

Тобто, як свідчить вираз (2.4) та рис. 2.4, коли значення величин b_i та d_ω співпадає, біт b_i контейнеру не зазнає змін.

При цьому, на рис. 2.4 показано процес вбудовування I біт. Даний сценарій може існувати в умовах, коли кількість I біт, що інкапсулюються, не перевищують потенційний V обсяг контейнеру. Тобто, формальна умова гарантованого внесення цифрової мітки/прихованого повідомлення наступна:

$$V \geq I. \tag{2.5}$$

2.3.2 Інкапсуляції даних за принципу фронтального вбудовування з довільними точками входу

Даний принцип є, у сутності, варіацією фронтального підходу, за винятком того, що [13]:

- інкапсуляція починається з μ,υ -го біту контейнеру;

- єдиний простір Λ' заповнення у контейнері Λ може утворюватися поєднанням ряду Ξ підпросторів Λ'_ξ, тобто:

$$\Lambda' = \bigcup_{\xi=1}^{\Xi} \Lambda'_\xi . \tag{2.6}$$

Відповідно, для цього випадку у процесі як інкапсуляції, так і зчитування вбудованих даних, на відміну від класичного фронтального способу, необхідно використовувати стегоключ $К$, який, у свою чергу, матиме наступний формат:

$$К = \{\Xi, \text{dir}, (b_{\mu,\upsilon}^{(\text{strt}1)}; b_{\mu,\upsilon}^{(\text{end}1)}), ..., (b_{\mu,\upsilon}^{(\text{strt}\xi)}; b_{\mu,\upsilon}^{(\text{end}\xi)}), ..., (b_{\mu,\upsilon}^{(\text{strt}\Xi)}; b_{\mu,\upsilon}^{(\text{end}\Xi)})\}, \tag{2.7}$$

де dir - параметр, що встановлює напрямок вбудовувань згідно з принципом (2.4) – за рядками, чи за стовпцями;

Ξ - параметр, що вказує на те, скільки фрагментів заповнення міститиме контейнер;

$(b_{\mu,\upsilon}^{(\text{strt}\xi)}; b_{\mu,\upsilon}^{(\text{end}\xi)})$ - початкова та кінцева координата ξ-ї зони вбудовування відповідно.

У свою чергу, величину Ξ та координати $(b_{\mu,\upsilon}^{(\text{strt}\xi)}; b_{\mu,\upsilon}^{(\text{end}\xi)})$ для кожної з зон не може бути обрано довільно. Тут слід брати до уваги обмеження (2.1), зокрема забезпечити виконання умов:

$$V_m = \sum_{\xi=1}^{\Xi} B_\xi \le 0{,}6V_c , \tag{2.8}$$

де B_ξ - об’єм біт, ξ-ї зони вбудовування.

Далі розглянемо, за яких умов та які алгоритми базуються на використанні фронтальних підходів до заповнення контейнерів

2.4 Опосередкована інкапсуляція даних на базі використання значень величин компонент трансформованих блоків

Даний метод використовується для розміщення прихованих даних у межах jpeg-контейнерів [1, 4, 7].

У рамках цього методу до трансформованого блоку $\hat{\beta}(ch)_{x,y}$ може бути внесено 1 біт. Такий підхід, у свою чергу, потенційно здатен забезпечити високий ступінь захищеності вбудованої інформації, так як незначна наповненість контейнеру, 1 біт з 64 у межах блоку $\hat{\beta}(ch)_{x,y}$, не дозволяє ефективно використовувати переважну більшість уніфікований методів стеганографічного аналізу на базі виявлення статистичних аномалій контейнерів.

Також безумовними перевагами методу є:

- простота та тривіальність реалізації;

- непрямий спосіб інкапсулювання біт, при якому у контейнер вбудовується щонайменше 1 біт d_ω приховуваних даних;

- відсутність типових ознак факту присутності інкапсуляції, що характерні багатьом поширеним методам; зокрема, не зазнають змін як статистичні характеристики блоків контейнеру, так і зони ймовірного вбудовування, зокрема - НЗБ.

Традиційно для методів, орієнтованих на контейнери jpeg, етап інкапсулювання біт секретного повідомлення реалізується по завершенню технологічного кроку ортогонального ДКП-перетворення, квантизації утворених компонент та наступного їх округлення (рис.2.3).

При цьому на той випадок, що у межах щойно перетвореного блоку $\hat{\beta}(ch)_{x,y}$ виявляється деяка різниця абсолютних значень компонент $\eta(ch)_{x,y}$, що перевищує деякий попередньо встановлений поріг ε, такий блок буде задіяно для інкапсуляції символу 0.

З іншого боку, якщо величина різниці абсолютних величин $\eta(ch)_{x,y}$ буде не більшою, ніж деякий поріг ε, такий блок міститиме у собі надалі символ 1.

Даний принцип інкапсуляції ілюструє наступний вираз:

$$\begin{cases} \left|\eta(ch)_{x,y}^{(max)}\right| - \left|\eta(ch)_{x,y}^{(min)}\right| > \varepsilon \rightarrow d_\omega = 0; \\ \left|\eta(ch)_{x,y}^{(max)}\right| - \left|\eta(ch)_{x,y}^{(min)}\right| < \varepsilon \rightarrow d_\omega = 1, \end{cases} \quad (2.9)$$

де $\eta(ch)_{x,y}^{(max)}$ та $\eta(ch)_{x,y}^{(min)}$ - відповідно, величини максимального та, мінімального значень компонент $\eta(ch)_{x,y}$ у межах блоку $\hat{\beta}(ch)_{x,y}$ що належить каналу ch.

Сам алгоритм містить у собі такі технологічні кроки, як:

1. Зчитування біту d_i повідомлення, яке вбудовується.
2. Вимірювання різниць між $\left|\eta(ch)_{x,y}^{(max)}\right|$ та $\left|\eta(ch)_{x,y}^{(min)}\right|$.
3. Порівняння величин d_ω та ε.
4. Прийняття рішення щодо необхідності корегування значень $\left|\eta(ch)_{x,y}^{(max)}\right|$ та $\left|\eta(ch)_{x,y}^{(min)}\right|$ та, безпосередньо, саме корегування.
5. Повтор пунктів 1-4 кількість разів, необхідну для внесення усього повідомлення.

При цьому, корегування величин $\eta(ch)_{x,y}^{(max)}$ та $\eta(ch)_{x,y}^{(min)}$ за пунктом 4 алгоритму передбачає їх модифікацію на величину $\Delta\eta$, як показує наступний вираз:

$$\Delta\eta = \varepsilon - \left|\eta(ch)_{x,y}^{(max)}\right| - \left|\eta(ch)_{x,y}^{(min)}\right|. \quad (2.10)$$

Тут можуть, розглядатися наступні випадки, зокрема:

- модифікується або значення $\eta(ch)_{x,y}^{(max)}$ або $\eta(ch)_{x,y}^{(min)}$;
- зміні підлягає як величина $\eta(ch)_{x,y}^{(max)}$ так і $\eta(ch)_{x,y}^{(min)}$.

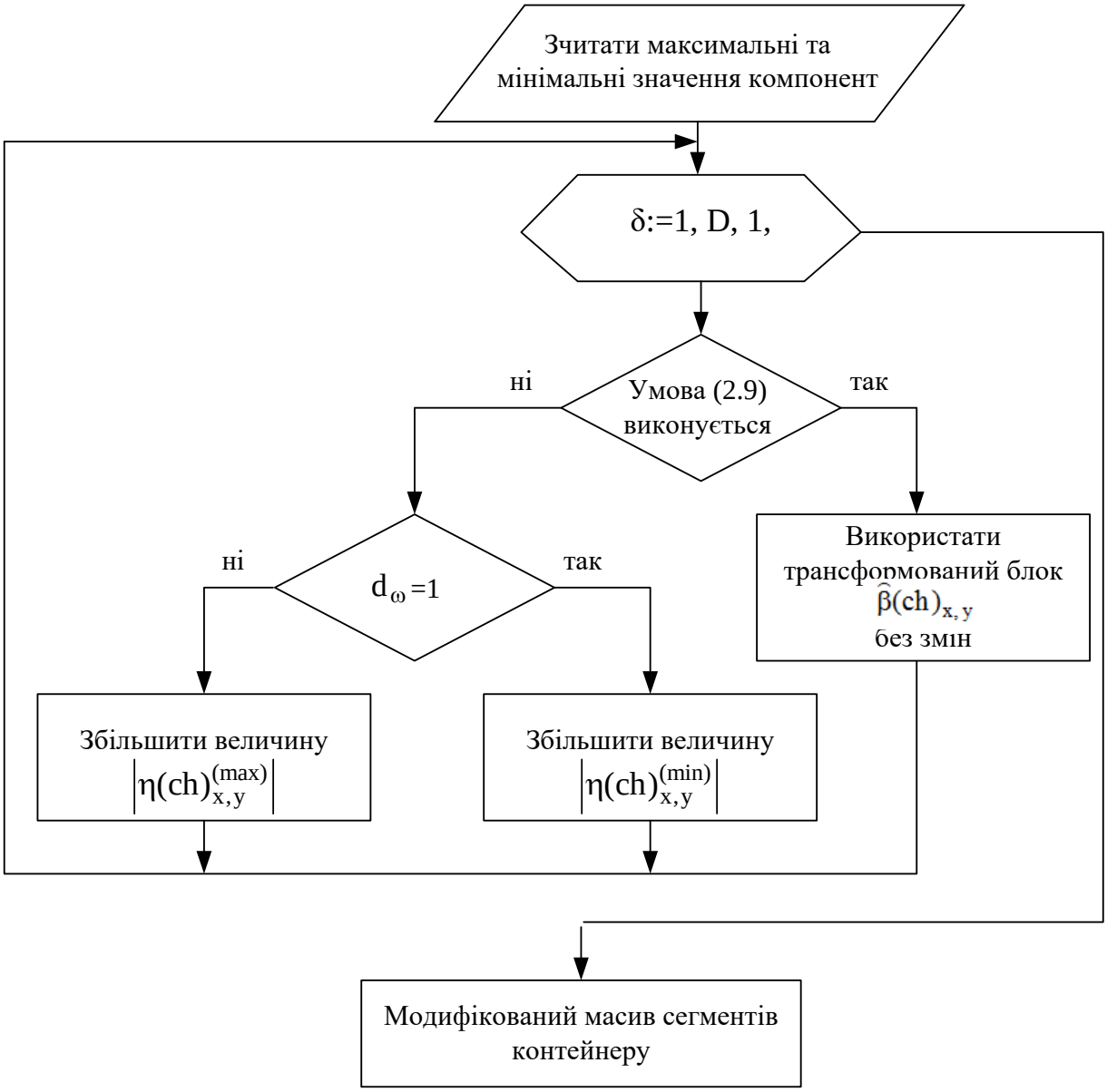


Рисунок 2.5 - Загальний принцип функціонування алгоритму інкапсуляції даних на базі використання значень величин компонент трансформованих блоків

Отже, як видно з аналізу алгоритму, попри відсутність прямої модифікації даних на рівні біт, за результатами його виконання контейнер зазнає часткового викривлення.

Зрозуміло, що це, у свою чергу, зумовлюється довільним розподілом як характеристик компонент $\eta(ch)_{x,y}$ усередині блоків $\widehat{\beta}(ch)_{x,y}$, так і випадковим змістом повідомлення, яке потребує вбудовування. Це може вважатися

суттєвим недоліком методу, так як результат модифікації величин $\left|\eta(ch)_{x,y}^{(max)}\right|$ та $\left|\eta(ch)_{x,y}^{(min)}\right|$ за певних умов є візуально помітним (рис. 2.6).

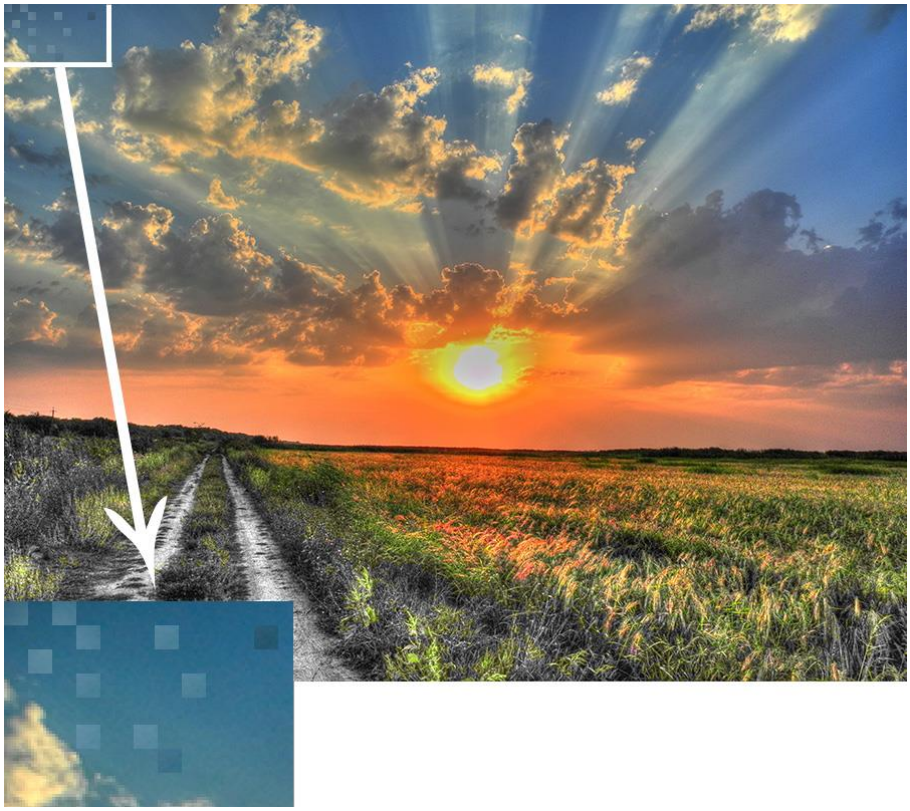


Рисунок 2.6 – Приклад візуального викривлення контейнеру

У таких умовах для збільшення ефективності методу може бути використано один з наступних підходів, а саме [1, 4, 7]:

- вибір у якості контейнерів такі зображення, що містять у собі велику кількість контурів, тобто, характеризуються візуальною складністю; на цей випадок викривлення, що вносяться у контейнер, теоретично може бути нівельовано;

- використання удосконаленої версії методу, - т.з. метод Langelaar.

При цьому, на випадок застосування методу інкапсуляція даних на базі використання значень величин компонент трансформованих блоків для передавання секретних повідомлень (тобто, рішення класичного завдання стеганографії) перший підхід має сенс. У нашому ж випадку, коли мова йде про внесення ЦВЗ, контейнер обиратися довільно не може.

Отже, доцільним є використання удосконаленої версії методу.

Так, метод Langelaar, як і розглянута версія методу, орієнтована на роботу з компонентами $\eta(ch)_{x,y}$, отриманими після ДКП на рівні блоків $\hat{\beta}(ch)_{x,y}$. Він містить у собі такі технологічні етапи, як:

1. Створення бінарної маски на базі простого псевдовипадкового алгоритму вигляду $\mathrm{rand}(x,y) \in \{0,1\}$.

2. Ділення початкового блоку $\hat{\beta}(ch)_{x,y}$ на 2 частини, тобтоо - $\hat{\beta}(ch)^{(1)}_{x,y}$ та $\hat{\beta}(ch)^{(2)}_{x,y}$ у відповідності до утвореної маски.

3. Розрахунок середньої величини яскравості для складника $\hat{\beta}(ch)^{(1)}_{x,y}$ блоку згідно з наступним виразом:

$$\overline{\eta}(Y)_{x,y} = \frac{\sum_{x=1}^{8-\nu}\sum_{y=1}^{8-\theta}\eta(Y)_{x,y}}{(8-\nu)(8-\theta)}, \qquad (2.11)$$

де ν та θ – значення, на які вихідний блок $\hat{\beta}(ch)_{x,y}$ є більшим за розмір складника $\hat{\beta}(ch)^{(1)}_{x,y}$.

Таким же чином розраховується середня величина яскравості для випадку складника $\hat{\beta}(ch)^{(2)}_{x,y}$, тобто:

$$\overline{\eta}(Y)_{x,y} = \frac{\sum_{x=1}^{\nu}\sum_{y=1}^{\theta}\eta(Y)_{x,y}}{\nu\theta}. \qquad (2.12)$$

4. Задання величини порогу Φ.

5. Вбудовування біта d_ω секретного повідомлення керуючись принципом, який подано наступним виразом:

$$d_\omega = \begin{cases} 1, \mid \hat{\beta}(ch)^{(1)}_{x,y} - \hat{\beta}(ch)^{(2)}_{x,y} > \Phi, \\ 0, \mid \hat{\beta}(ch)^{(1)}_{x,y} - \hat{\beta}(ch)^{(2)}_{x,y} < -\Phi. \end{cases} \qquad (2.13)$$

Далі у випадку, коли система нерівностей (2.13), не виконується, використовується корегування величин компонент яскравості складника $\widehat{\beta}(ch)^{(2)}_{x,y}$.

У свою чергу, зчитування біту d_ω, вбудованого раніше, на боці приймача попередньо потребує обчислення середніх значень компонент яскравості у межах складників $\widehat{\beta}(ch)^{(1)}_{x,y}$ та $\widehat{\beta}(ch)^{(2)}_{x,y}$ початкового блоку. Після цього процедура зчитування біта d_i може бути реалізована відповідно до співвідношення:

$$d_\omega = \begin{cases} 1, \mid \widehat{\beta}(ch)^{(1)}_{x,y} - \widehat{\beta}(ch)^{(2)}_{x,y} > 0, \\ 0, \mid \widehat{\beta}(ch)^{(1)}_{x,y} - \widehat{\beta}(ch)^{(2)}_{x,y} < 0. \end{cases} \quad (2.14)$$

Дані, інкапсульовані таким чином до графічного контейнеру, у загальному випадку характеризуються високою ступінню захищеності та високою складністю виявлення.

Разом з тим, ключовим недоліком розглянутого методу Langelaar є низька робастність, тобто, стійкість до перетворень. Даний фактор, у свою чергу, суттєвим чином обмежує діапазон застосування даного методу.

При цьому, ефективність даного методу, як і передуючого йому методу опосередкованої інкапсуляції даних на базі використання значень величин компонент трансформованих блоків, також суттєво залежить від особливостей застосовуваних контейнерів.

Як і для попередньо розглянутого методу, можуть спостерігатися спотворення контейнерів (рис.2.6), що також обмежує його застосовуваність.

У свою чергу, недоліків, характерних розглянутим методам, позбавлений метод модифікації найменш значущих біт, або LSB (Less Significant Bit).

2.5 Метод LSB-модифікації

Найчастіше фронтальний спосіб заповнення застосовується на випадок, коли контейнером є графічний об’єкт, тобто, що відповідає умовам, зазначеним у технічному завданні. За цих умов вираз (2.4) може бути деталізовано до наступного вигляду [1-3, 5, 13]:

$$\begin{cases} b(ch)_{k\ell} := \neg b(ch)_{k\ell} \mid b(ch)_{k\ell} \neq d_{\omega}; \\ b(ch)_{k\ell} := b(ch)_{k\ell} \mid b(ch)_{k\ell} = d_{\omega}, \end{cases} \tag{2.15}$$

де $b(ch)_{k\ell}$ - біт контейнеру з координатами $(k;\ell)$ каналу ch. Каналом тут може бути або одна з множин простору RGB, або – YUV (YCbCr).

За означеним принципом сьогодні побудовано класичний метод LSB-стеганографії. Його сутність полягає у модифікації молодших (LSB – least significant bat) біт контейнеру відповідно до закону вбудовуваних даних. Така можливість зумовлюється тим, що викривлення, внесені таким чином на рівні множини біт LSB є візуально непомітними.

Згідно засад LSB, відносно початкового контейнеру послідовно виконується ряд операцій, а саме [11]:

1. Зміна формату представлення. Так, контейнер, що у вихідному вигляді являє собою матрицю $(M \times N)$ десяткових елементів, підлягає переведенню до бінарного формату. При цьому, на випадок bmp формат конвертується на рівні пікселів $p_{x,y}$, а для jpeg – на рівні компонент $\eta_{x,y}$, як показано наступним виразом:

$$p_{x,y}, \eta_{x,y} = \sum_{\theta=7}^{0} b_{\theta}(ch)_{x,y} \times 2^{(\theta-1)}, \tag{2.16}$$

де $b_{\theta}(ch)_{x,y}$ - біт θ-го розряду відповідного каналу.

Зазначимо, що у випадку обробки компонент $\eta_{x,y}$ можливі значення параметру ch є такими:

- $ch = 1$ - для каналу Y яскравості;
- $ch = 2$ - для каналу Cb;
- $ch = 3$ - для каналу Cr.

У свою чергу, для пікселів $p_{x,y}$ відповідно маємо:

- $ch = 1$ - для каналу R;
- $ch = 2$ - для каналу G;
- $ch = 3$ - для каналу B.

2. Утворення множин Ψ_{θ} біт одного розряду, що здійснюється на основі принципу, поданого виразом далі:

$$\Xi(\theta) = \bigcup_{x=1}^{H}\bigcup_{y=1}^{W} b_{\theta}(ch)_{x,y} \qquad (2.17)$$

За ідеології LSB-методу, у межах графічного контейнеру модифікація біт за виразом (2.15) здійснюється на рівні $\theta = 0$, тобто, утворюється множина Ψ_0 бінарних елементів молодшого розряду розмірністю $M \times N$, що відповідає розміру початковому файлу. Таким чином, теоретично можлива ємність V одного контейнеру на базі LSB з фронтальним заповненням може сягати $M \times N$ біт в умовах, коли задіяно єдиний канал і $3 \times M \times N$ - за умови максимального завантаження. Дане твердження є справедливим як для bmp, так і для jpeg контейнерів [13].

Разом з тим, як свідчить властивість (1.3) контейнеру, ріст величини V незмінно веде до падіння рівня захищеності P стеганограми [14].

3. Фронтальний обхід контейнеру за напрямком рядків, або стовпців, починаючи з деякої $(\alpha;\beta)$ координати. Зазвичай для інкапсуляції прихованих повідомлень, які надалі буде передано у мережу, звичайними є наступні ситуації, зокрема:

- $\alpha = 1$ і $\beta = 1$ (вбудовування з початку координат у напрямку або рядків, або за стовпцями);

- $\alpha = 1$ і $\beta \in [2; N]$ (вбудовування виконується за стовпцями з довільної позиції у зазначеному діапазоні);

$\alpha \in [2; M]$ і $\beta = 1$ (вбудовування виконується за рядками з довільної позиції у зазначеному діапазоні).

У свою чергу, для нанесення цифрових міток характерною є ситуація, коли $(\alpha \in [1; M]) \,\&\, (\beta \in [1; N])$.

2.6 Недоліки методів, що базуються на фронтальному заповненні контейнеру

Як попередньо розглянуті методи, орієнтовані на непряме вбудовування даних з урахуванням величин найвищого $\left|\eta(ch)_{x,y}^{(max)}\right|$ та найменшого $\left|\eta(ch)_{x,y}^{(min)}\right|$ значень компонент в одному з каналів, так і LSB-орієнтовані методи у загальному випадку передбачають єдиний погляд на вибір локацій у середині контейнеру для реалізації інкапсулювання. При цьому, одному біту d_{ω}

повідомлення ($\omega = \overline{1;\Omega}$), яке вбудовується, ставиться у відповідність деяка структурна одиниця контейнеру, на рівні якого виконується внесення даних [8].

Так, для методів інкапсуляції даних на базі використання значень величин компонент трансформованих блоків це, безпосередньо, самі блоки $\hat{\beta}(ch)_{x,y}^{(2)}$, у межах кожного з яких за потреби можуть корегуватися значення $\left|\eta(ch)_{x,y}^{(max)}\right|$ та $\left|\eta(ch)_{x,y}^{(min)}\right|$, тоді як для LSB це – біти $b_0(ch)_{x,y}$ нульового розряду. При цьому, у загальному випадку застосовується фронтальний сценарій заповнення, що включає у себе такі кроки, як [7]:

1. Зчитати дані структурної $S_{i,j}$ ($i = \overline{1;\Phi}$, $j = \overline{1;\Theta}$) одиниці контейнеру Λ_k на позиції (1,1) контейнеру, тобто, $S_{1,1}$ (рис. 2.7).

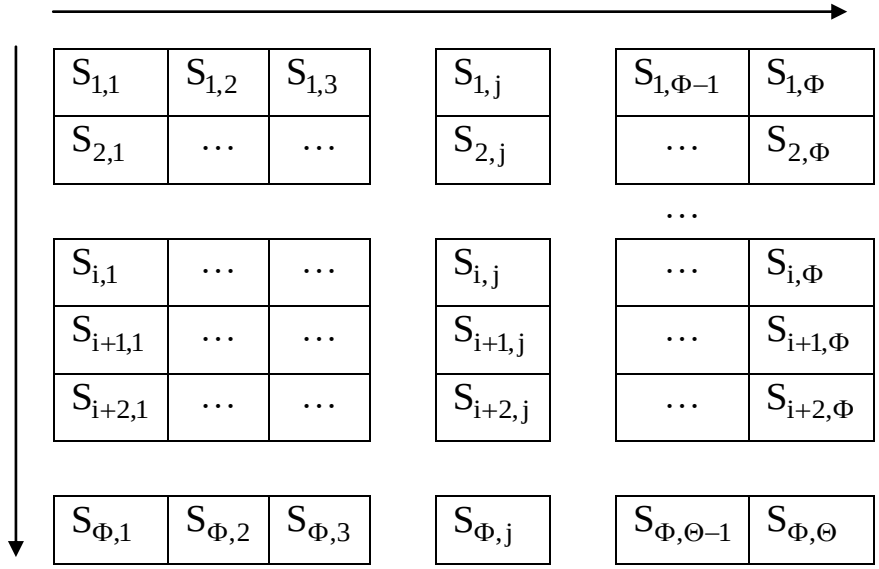


Рисунок 2.7 – Матриця структурних одиниць контейнеру та порядки його обходу у випадку фронтального заповнення

2. Зчитати d_ω.

3. Перевірити умови необхідності зміни змісту поточної структурної одиниці (вирази 2.13-2.15) та за потреби внести такі зміни.

4. Зміститися до наступної структурної одиниці, тобто, $S_{1,2}$ (обробка за рядками) чи $S_{2,1}$ (обробка за стовпцями).

5. Повторити кроки 1-3 до моменту, коли буде досягнуто останньої структурної одиниці у рядку (стовпці), тобто, виконується умова:

$$(i = \Phi) \vee (j = \Theta). \tag{2.18}$$

6. Зміститися до наступного рядка (стовпцю), тобто:

$$\begin{cases} j := j+1 | i = \Phi; \\ i := i+1 | j = \Theta. \end{cases} \tag{2.19}$$

7. Повторити кроки 1-5 до виконання умови:

$$\omega = \Omega, \tag{2.20}$$

що сигналізує про те, що останній біт d_Ω повідомлення вбудовано, або умови:

$$((i = \Phi) \,\&\, (j = \Theta)) \,|\, \omega \neq \Omega, \tag{2.21}$$

тобто, для інкапсуляції використано усі доступні біти контейнеру Λ_k, при цьому його ємності недостатньо для вміщення поточного повідомлення.

Таким чином, якщо умова (2.21) виконується, далі приймається рішення про необхідність залучення додаткового контейнеру Λ_{k+1}

Попри простоту та легкість реалізації, розглянутим методам властивий ряд суттєвих недоліків, що обмежують можливості їх повсякденного використання.

Так, класичний метод LSB, реалізований на базі графічних контейнерів, характеризується низьким рівнем захищеності P. Зокрема, для викриття модифікацій достатньо виконати декомпозицію компонент/пікселів контейнеру за виразом (2.16) та побудувати матрицю LSB (вираз 2.17), що демонструється рис.2.8 [12, 14, 15].

Рисунок 2.8 – Ознаки інкапсуляції даних, виявлені на рівні LSB контейнеру-зображення

У свою чергу, на випадок методів інкапсуляції даних на базі використання значень величин компонент трансформованих блоків викривлення можуть бути візуально видимі без будь-якої додаткової обробки [14].

Це свідчить про неефективність фронтального підходу для реалізації розглянутих методів стеганографічного вбудовування. Означені недоліки, хоч і меншою мірою, також властиві також для фронтального вбудовування з довільними точками входу. У зв’язку з цим, пропонується розглянути альтернативні алгоритми реалізації методів вбудовування даних, зокрема, ЦВЗ.

3. ПОБУДОВА МЕХАНІЗМУ ВНЕСЕННЯ ЦВЗ ДЛЯ СТАТИЧНИХ ОБ'ЄКТІВ БЕЗ ФРОНТАЛЬНОГО ЗАПОВНЕННЯ

3.1 Вибір структурних одиниць контейнеру, на рівні яких реалізується механізм інкапсуляції

У загальному випадку прикладом статичного об'єкту є графічний контейнер. Тут вибір рівня структурних одиниць для побудови альтернативного способу заповнення контейнеру виконується виходячи з наступних міркувань:

- розмір ЦВЗ може бути теоретично довільним, тобто, має досягатися достатній рівень ємності V за умови забезпечення високого рівня захищеності, як показано умовами 1.1, 1.3 та 1.4 [1, 8, 14];

- спосіб вибору структурних одиниць контейнеру для подальшої інкапсуляції даних має ураховувати недоліки, виявлені для випадку фронтального підходу;

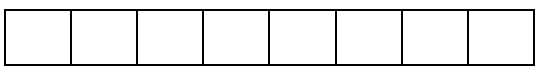
- має забезпечуватися захищеність вбудованих даних та їх стійкість до стандартизованих методів виявлення.

Отже, у якості структурної одиниці, асоційованої з одним символом d_ω повідомлення, у зазначених вище умовах недоцільно розглядати блок $\hat{\beta}(ch)_{x,y}$, так як при цьому ємність V контейнеру може бути надто низькою. Відтак, у цьому випадку асоційованою структурною одиницею буде виступати біт $b(ch)_{x,y}$.

У свою чергу, оскільки ряд алгоритмів стегоаналізу, орієнтованих на обробку графічних контейнерів, фокусується на рівні блоків $\hat{\beta}(ch)_{x,y}$, як окремих суб-просторів вихідного контейнеру, де здійснюються базові та доаткові локальні перетворення, має сенс змінити стандаризований розмір блоку.

При цьому, замість одного блоку $\hat{\beta}(ch)_{x,y}$ що має розмір 8x8, надалі розглядатиметься множина {B} спліт-блоків $\hat{\beta}(ch)_{x,y}^{(k.\ell)}$, $k,\ell=\overline{1,2}$.

Кожен з таких спліт-блоків складає 1/4 початкового блоку, тобто, його розмірність становить 4x4 (рис.3.1).



$\hat{\beta}(ch)_{x,y}$

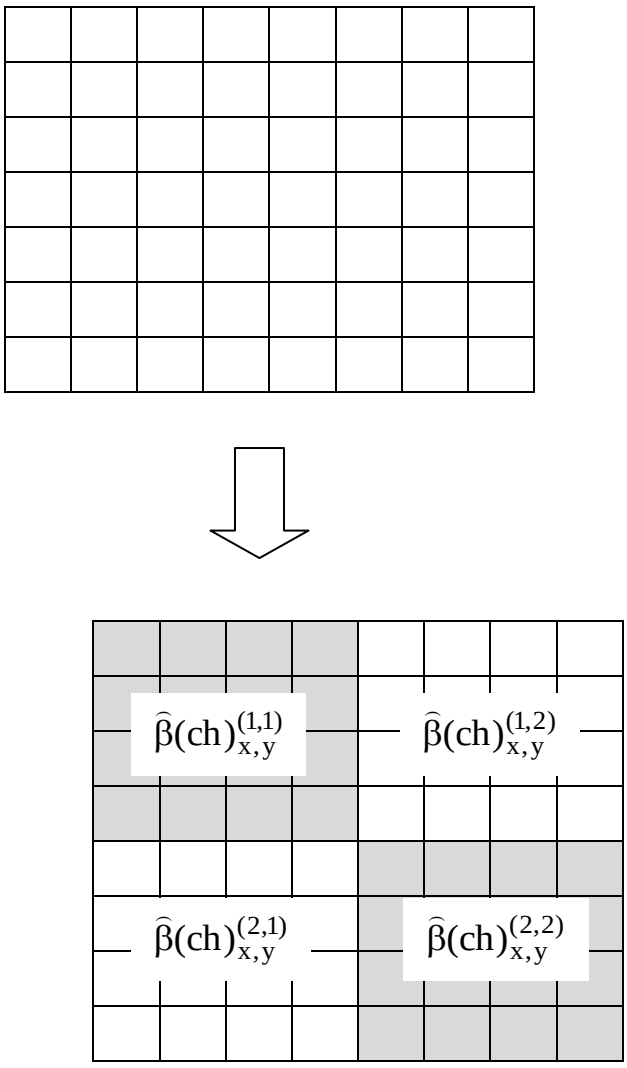


Рисунок 3.1 – Поділ вихідного блоку $\widehat{\beta}(ch)_{x,y}$ на сукупність спліт-блоків $\widehat{\beta}(ch)^{(k.\ell)}_{x,y}$

3.2 Принцип інкапсуляції даних на рівні спліт-блоків

Після того, як сукупність спліт-блоків $\widehat{\beta}(ch)^{(k.\ell)}_{x,y}$ сформовано, серед них виявляється один або кілька, що суттєво відрізняються від інших за одним, або рядом статистичних показників.

Наприклад, таким показником може бути визначена кількість ϑ бінарних переходів з 0 на 1, як така, для якої справедливо:

$$\vartheta \to \vartheta(\widehat{\beta}(ch)_{x,y})_{max}. \qquad (3.1)$$

Теоретично незалежно від інших у блоці $\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}}$ використовуватися можуть усі спліт-блоки $\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}}^{(\mathrm{k}.\ell)}$.

Разом з тим, для збільшення захищеності алгоритму у цілому доцільно застосовувати одночасно не більш, ніж 2 спліт-блока, які надалі позначаимуться як $\widehat{\beta}_1(\mathrm{ch})_{\mathrm{x,y}}^{(\mathrm{k}.\ell)}$ та $\widehat{\beta}_2(\mathrm{ch})_{\mathrm{x,y}}^{(\mathrm{k}.\ell)}$. У такому разі у рамках $\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}}$ визначення першого та другого спліт-боків виконуватиметься за системою виразів:

$$\begin{cases} \widehat{\beta}_1(\mathrm{ch})_{\mathrm{x,y}}^{(\mathrm{k}.\ell)} = \widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}}^{(\mathrm{k}.\ell)} \mid \vartheta(\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}}^{(\mathrm{k}.\ell)}) = \vartheta(\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}}^{(\mathrm{k}.\ell)})_{\max}; \\ \widehat{\beta}_2(\mathrm{ch})_{\mathrm{x,y}}^{(\mathrm{k}.\ell)} = \widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}}^{(\mathrm{k}.\ell)} \mid \vartheta(\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}}^{(\mathrm{k}.\ell)})\ \vartheta \in (\vartheta(\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}})_{\max}; \alpha\vartheta(\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}})_{\max}]. \end{cases} \quad (3.2)$$

Тобто, перший спліт-блок – той, якому відповідає $\vartheta(\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}})_{\max}$, відповідно, другий – для якого величина ϑ знаходиться у діапазоні $(\vartheta(\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}})_{\max}; \alpha\vartheta(\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}})_{\max}]$.

За замовчуванням приймається величина $\alpha = 0{,}7$. Надалі дана величина може бути скорегована.

Стегоключ для режима інкапсулювання, у ході якого для вбудовування даних задіюється 1 спліт-блок $\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}}^{(\mathrm{k}.\ell)}$, якому якому відповідає $\vartheta(\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}})_{\max}$, у першому полі R містить нульовий елемент (в інакшому випадку - 1).

Для вбудовування на рівні одного (або двох) спліт-блоків блоків може бути задіяно або 1, або 2 рядки (стовпця). При цьому, якщо інкапсуляція виконується за стовпцями, у наступному полі σ стегоключа вноситься 0, рядку - 1.

Для спліт-блоку 2 рядки (стовпця) будуть використовуватися за умови, що:

$$\vartheta \geq \vartheta_{\mathrm{th}}, \quad (3.3)$$

де p_{th} - деяке порогове значення, встановлене раніше (при цьому зрозуміло, що $\vartheta \in [0;16]$).

У свою чергу, номер n_1 першого рядку (стовпцю) у межах тих спліт-блоків $\widehat{\beta}(\mathrm{ch})_{\mathrm{x,y}}^{(\mathrm{k}.\ell)}$, що надалі буде використані у ході інкапсуляції біт секретного

повідомлення, за умови, що таких рядків буде 2 (та єдиного для випадку, коли використовуватиметься один рядок/стовпець), може бути визначений за наступною системою виразів:

$$\begin{cases} n=1 \mid \sum \upsilon = \varphi; \\ n=2 \mid \sum \upsilon \in (\varphi; \varphi\phi_1]; \\ n=3 \mid \sum \upsilon \in (\varphi\phi_1; \varphi\phi_2]; \\ n=4 \mid \sum \upsilon \in (\varphi\phi_2; \varphi\phi_3], \end{cases} \tag{3.4}$$

де υ - кількість одиничних (нульових) елементів, виявлених у межах спліт-блоку;

$\phi_1 ... \phi_3 \in (0;1)$ - умовні множники, кожен з яких є елементом стегоключа;

φ - порогова величина, що є елементом стегоключа.

При цьому, вибір номеру n_2 другого рядка/стовпця, (якщо його використання передбачається налаштуванням режиму роботи) виконується на базі наступного виразу:

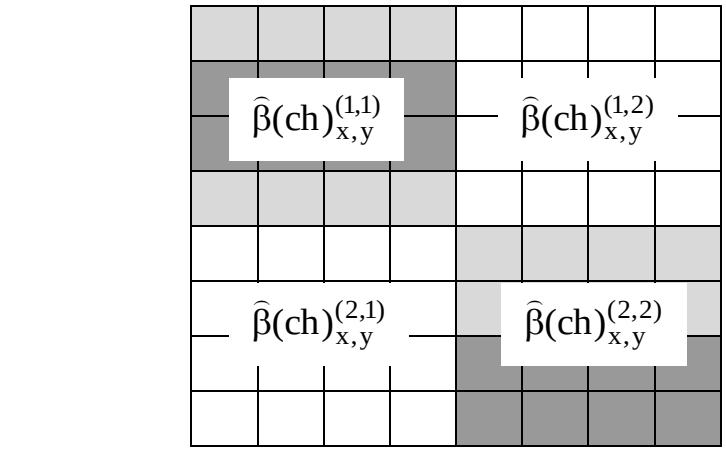
$$\begin{cases} n_2 = n_1 + 1 \mid n_2 = \overline{2;3}; \\ n_2 = n_1 - 1 \mid n_2 = \overline{3;4}, \end{cases} \tag{3.5}$$

У цьому випадку, по-перше, уникаються випадки невизначеності, пов’язані з ймовірним переповненням допустимої розмірності величини n, що, у сутності, обмежена діапазоном $n = \overline{1;4}$. По-друге, забезпечується можливість однозначного розпізнання задіяних рядків/стовпців у ході процедури зчитування інкапсульованих даних.

Вигляд ключа K для даного випадку буде наступним:

$$K = \{R; \sigma; \phi_1; \phi_2; \phi_3; \varphi\}. \tag{3.6}$$

Приклад заповнення блоку $\hat{\beta}(ch)_{x,y}$ на випадок використання 2 рядків/стовпців для інкапсуляції даних у рамках попередньо обраних спліт-блоків $\hat{\beta}(ch)_{x,y}^{(k.\ell)}$ ілюструє рис. 3.2.



- рядки, у які вбудовуються приховані дані

Рисунок 3.2 – Приклад заповненого блоку $\widehat{\beta}(ch)_{x,y}$ за умови використання 2 рядків

3.3 Критичний аналіз дослідженого підходу до інкапсуляції даних на засадах LSB та його доопрацювання

Як показує огляд принципу функціонування методу, що базується на використанні спліт-блоків у рамках вихідного блоку $\widehat{\beta}(ch)_{x,y}$, з формальної точки зору його реалізація позбавлена недоліків, притаманних фронтальним способам заповнення [8, 12, 14].

Спочатку виконаємо розрахунок діапазону можливої заповненості контейнеру.

3.3.1 Оцінка мінімального та максимального рівня наповненості контейнеру у досліджуваних умовах

Згідно засад LSB інкапсуляція біт приховуваного повідомлення, у т.ч. ЦВЗ, жодним чином не спричинює виникнення суттєвих викривлень. Виходячи з цього, дані можуть вбудовуватися як у яскравісні $\widehat{\beta}(Y)_{x,y}$, так і у хроматичні блоки - $\widehat{\beta}(Cb)_{x,y}$ та $\widehat{\beta}(Cr)_{x,y}$ [9-13, 16-18].

Разом з тим, залежно від обраного рівня колірної субдискретизації, при незмінній розмірності $\widehat{\beta}(Y)_{x,y}$ розмірність блоків $\widehat{\beta}(Cb)_{x,y}$ та $\widehat{\beta}(Cr)_{x,y}$ може варіюватися. Зокрема, у режимі (4:1:1) їх розмір дорівнюватиме 4x4. За таких

умов у межах кожного з них може існувати виключно 1 спліт-блок $\widehat{\beta}(\mathrm{Cb})_{x,y}^{(k.\ell)}$ ($\widehat{\beta}(\mathrm{Cr})_{x,y}^{(k.\ell)}$).

При цьому, за умови мінімального навантаження у межах спліт-блоків $\widehat{\beta}(\mathrm{Cb})_{x,y}^{(k.\ell)}$ ($\widehat{\beta}(\mathrm{Cr})_{x,y}^{(k.\ell)}$) заповненню підлягає виключно 1 рядок/стовпець. Отже, з урахуванням цього у просторі Cb чи Cr заповненню за найгірших умов теоретично може підлягати $\frac{1}{4}$ усього вихідного об'єму (мається на увазі, що використовується один хроматичний канал).

Тобто, якщо один вихідний блок $\beta_{x,y}$ розглядати як:

$$\beta_{x,y} = \beta(Y)_{x,y} \,\&\, \beta(Cb)_{x,y} \,\&\, \beta(Cr)_{x,y}, \tag{3.7}$$

то на рівні LSB, таким чином, будуть присутні 4 спліт-блоки $\widehat{\beta}(Y)_{x,y}^{(k.\ell)}$, що будуть незаповненими, 1 незаповний та заповнений хроматичні спліт-блоки. Отже, загальна кількість спліт-блоків 4x4 – 6. З них заповненими у межах одного рядку ($\frac{1}{4}$ обсягу спліт-блоку) є лише один. Таким чином, загальна кількість біт LSB у режимі (4:1:1) буде дорівнювати $V_c = 64+16+16 = 96$. Водночас, один хроматичний спліт-блок міститиме 4 біти. Тобто, навіть у просторі LSB при цьому відношення V_m до V_c дорівнюватиме $\frac{V_m}{V_c} = \frac{4}{96} \approx 0{,}04$, що є значно нижче, ніж зазначено умовою (2.1), коли факт існування інкапсуляції може бути викрито статистичними методами стегоаналізу ($\frac{V_m}{V_c} \geq 0{,}6$).

Тепер розглянемо випадок найвищого заповнення блоку $\beta_{x,y}$, що відповідає умовам, коли поле R стегоключа дорівнює 1 (використовуються 2 спліт-блоки), у рамках кожного з яких інкапсуляції підлягають 2 рядки для кожного з них. При цьому, за умов забезпечення найвищої заповненості, інкапсуляція виконуватиметься на рівні блоку $\widehat{\beta}(Y)_{x,y}$ яскравісних компонент (8x8). Тобто, для цього буде задіяно 4 рядки кожен довжиною 4 двійкових

символа. У свою чергу, для даного випадку $\frac{V_m}{V_c} = \frac{4x4}{96} \approx 0{,}16$, що також задовольняє вимогу (2.1).

Таким чином, навіть у режимі найвищої наповненості, розглянутий спосіб розміщення інкапсульованих даних формально можна вважати стійким до методів виявлення, що базуються на дослідженні статистичних аномалій ймовірного контейнеру.

У той же час, за певних умов контейнери, утворені у такий спосіб, можуть наслідувати властивості, характерні для фронтального способу заповнення. Зокрема, частковими випадками таких умов є [14, 15]:

$$\begin{aligned}&\left(n_1(B_1(\widehat{\beta}(ch)_{x,y})) = n_1(B_2(\widehat{\beta}(ch)_{x,y})) = \ldots = n_1(B_W(\widehat{\beta}(ch)_{x,y}))\right) \vee \\ &\vee \left(n_2(B_1(\widehat{\beta}(ch)_{x,y})) = n_2(B_2(\widehat{\beta}(ch)_{x,y})) = \ldots = n_2(B_W(\widehat{\beta}(ch)_{x,y}))\right),\end{aligned} \quad (3.8)$$

це свідчить про те, що у межах деякої кількості W сусідніх спліт-блоків для інкапсуляції використано рядки/стовпці з однаковими індексами.

Відповідно, у ході візуального аналізу LSB-змісту може біти виявлено ланцюжки фронтального заповнення, що суттєво знижує стійкість вбудованих даних.

Отже, ураховуючи виявлені закономірності, необхідно внести ряд додаткових обмежень до процесу заповнення LSB-простору.

3.3.2 Встановлення обмежень відносно реалізації процесу заповнення LSB-простору

Для того, щоб уникнути випадків зменшення стійкості інкапсульованих даних, викликаних, зокрема, умовами, як показує вираз (3.8), може бути реалізовано такі підходи, як [9, 13]:

- внесення обмеження на модифікацію біт для сусідніх у рядку/стовпцю блоків $\widehat{\beta}(ch)_{x,y}$ контейнеру;

- вибір для модифікації блоків $\widehat{\beta}(ch)_{x,y}$ у межах контейнеру, керуючись особливостями їх змісту.

Розглянемо більш детально кожен з підходів.

Нехай маємо контейнер розмірністю X на Y блоків $\hat{\beta}(ch)_{x,y}$, тобто, $x=\overline{1;X}$, та $y=\overline{1;Y}$. У рамках першого підходу після того, як деякий довільний блок $\hat{\beta}(ch)_{x,y}$ було модифіковано, інкапсуляцію може бути виконано виключно відносно блоків, що знаходяться на деякій фіксованій відстані Δx та/або Δy.

При цьому, відносно блоку $\hat{\beta}(ch)_{x,y}$ множина потенційно прийнятних блоків для подальшої ікапсуляції може бути описана на базі виразу [14]:

$$B' = \bigcup \hat{\beta}(ch)_{x\pm\Delta x, y\pm\Delta y}, \tag{3.9}$$

де B' - множина блоків, що підлягають модифікації.

На цей випадок стеганографічний ключ (3.6) буде розширено до вигляду:

$$K = \{R; \sigma; \phi_1; \phi_2; \phi_3; \varphi; \Delta x; \Delta y; r\}; \tag{3.10}$$
$$\Delta x, \Delta y = \overline{0; r},$$

де r - умовний поріг зміщення.

Разом з тим, очевидними недоліками такого підходу є надмірне ускладнення алгоритму вибору наступного блоку з тим, щоб забезпечити уникнення умов (3.8).

У той же час, більш доцільним для використання може алгоритм вибору блоків, що не орієнтується на фіксоване зміщення Δx та/або Δy відносно передуючого блоку.

Такий алгоритм передбачає включення до потенційної множини B' блоків $\hat{\beta}(ch)_{x,y}$ контейнеру, що у його межах характеризуються однаковою (або максимально схожі) за деякою інформативною ознакою.

Сенс такого підходу полягає у наступному:

1. У межах контейнеру Λ для кожного з блоків $\hat{\beta}(ch)_{x,y}$ виконується обчислення однієї чи кількох інформативних ознак. Це може бути одна зі статистичних характеристик блоку, виміряна як на рівні, власне, блоку $\beta_{x,y}$ у просторовій інтерпретації, так і на рівні блоку у спектральному поданні. Наприклад, для блоку $\hat{\beta}(ch)_{x,y}$ такою ознакою може розглядатися потужність абсолютних величин компонент, яка визнається як добуток P величин

компонент в одному з каналів, що розміщені на певних діагоналях (рис.3.3) [17-19]. Це еквівалентно виразу:

$$P = \prod_{\nu_c} \prod_{\nu_d} \eta(ch)_{x,y}^{(\nu_d,\nu_c)}, \qquad (3.11)$$

де ν_d - кількість діагоналей, задіяних у процесі обчислення інформативної ознаки;

ν_c - кількість компонент у межах однієї діагоналі.

Приклад на рис. 3.3 ілюструє умови обчислення величини P для випадку, коли $\nu_d = 2$.

1							
2	3						
	4	5					
		6	7				
			8	9			
				10	11		
					12	13	
						14	15

Рисунок 3.3 – Приклад вибору 2 діагоналей (4 та 10) для розрахунку інформативної ознаки потужність абсолютних величин компонент блоку

2. З блоків, інформативні ознаки яких відповідають попередньо встановленим критеріям, формується попередня множина B'. Для цього попередньо необхідно встановити допустимий діапазон величини P. З урахуванням цього, рішення про включення того чи іншого блоку $\hat{\beta}(ch)_{x,y}$ до множини B' прийматиметься на базі виразу:

$$\hat{\beta}(ch)_{x,y} \in B' \mid P(\hat{\beta}(ch)_{x,y}) \in [P_{max}; P_{min}], \qquad (3.12)$$

де P_{max} та P_{min} - межі діапазону допустимих значень P.

З урахуванням цього, зміст стеганографічного ключа буде змінено до вигляду:

$$К = \{R; \sigma; \phi_1; \phi_2; \phi_3; \varphi; diag; P_{max}; P_{min}\}, \quad (3.13)$$

де $diag$ - поле, що містить відомості про діагоналі, задіяні для розрахунку величини P.

Тут слід взяти до уваги той факт, що для реалізації розглядуваної концепції достатньо оперувати обмеженою кількістю діагоналей блоку, наприклад – 4. Оскільки за найгірших умов (коли розглядається блок $\hat{\beta}(Y)_{x,y}$) діагоналей у межах блоку – 15, відповідно, для розміщення одного індексу діагоналі достатньо 4 біт. Таким чином, для ідентифікаторів 4 діагоналей поле $diag$ має резервувати 2 байти (рис. 3.4).

Daig1 ID	Daig2 ID	Daig3 ID	Daig4 ID
1 байт		1 байт	

Рисунок 3.4 – Формат поля $diag$ стаганографічного ключа

Разом з тим, для уніфікації алгоритму необхідно передбачити можливість застосування довільної кількості діагоналей серед потенційно можливих 4-х. Для цього до стеганографічного ключа необхідно внести додаткове поле num_d, яке вказуватиме, скільки діагоналей для обчислення P використовується. Для поля num_d резервується 2 біти. З урахуванням цього, стегоключ матиме наступний формат:

$$К = \{R; \sigma; \phi_1; \phi_2; \phi_3; \varphi; num\_d; diag; P_{max}; P_{min}\}, \quad (3.14)$$

3. З множини B' виключаються такі блоки, $\hat{\beta}(ch)_{x,y}$, для яких виконується умова (3.8).

4. Встановлюється порядок обходу контейнеру на рівні блоків множини B' під час інкапсуляції/декапсуляції.

Говорячи про ЦВЗ, можемо зазначити, що порядок зчитування даних з контейнеру, як додатковий фактор захисту даних від зловмисника, може бути не таким важливим, як для випадку побудови стеганографічного каналу.

Разом з тим, для сприяння мінімізації ймовірності клонування ЦВЗ та подальшої дискредитації правовласника даний механізм також є суттєвим.

Розглянемо ряд підходів до побудови різних способів обходу контейнеру.

3.4 Додатковий механізм захисту інкапсульованих даних за рахунок маніпулювання напрямком обходу контейнера

Більшість стандартизованих алгоритмів зчитування даних графічного характеру дотримуються напрямку зростання індексів як компонент $\eta(ch)_{x,y}$ у межах блоку $\widehat{\beta}(ch)_{x,y}$, так і самих блоків $\widehat{\beta}(ch)_{x,y}$ у межах контейнеру Λ. Це у загальних рисах відповідає ідеології фронтального заповнення, як раніше зазначалося рис.2.7.

Якщо спосіб заповнення розглядати як додаткових елемент захисту – недоліки стандартизованого підходу очевидні [14, 15].

У свою чергу, до алгоритму встановлення порядку обходу множини B' може бути висунуто такі вимоги:

- простота побудови, тобто, додавання даного механізму до загального механізму обробки не повинно його суттєво ускладнювати;
- використання мінімально можливого рівня додаткової обчислювальної потужності;
- забезпечення можливості утруднення клонування ЦВЗ навіть за умови, що зловмиснику відомий його фрагмент.

У таких умовах з точки зору мінімізації ускладнення загального сценарію обробки, а також заощадження обчислювальної потужності доцільним може бути алгоритм обходу, який, як і спосіб вибору блоків $\widehat{\beta}(ch)_{x,y}$ для інкапсулювання, розглянутий у п. 3.3.2, базується на використанні статистичних властивостей блоків.

Зокрема, для уникнення великої кількості додаткових обчислень, пропонується скористатися попередньо розрахованими величинами P потужностей абсолютних величин компонент (вираз (3. 11)).

При цьому, оскільки для розгляду береться деякий діапазон $\Delta P = \overline{P_{max}; P_{min}}$, далі пропонується:

- встановити порядок обходу блоків множини B' виходячи з величин потужностей абсолютних величин компонент блоків $P(\hat{\beta}(ch)_{x,y})$;

- передбачити можливість зміни порядку обходу.

Припустимо, що у межах контейнеру Λ раніше виявлено g блоків $\hat{\beta}(ch)_{x,y}$, які може бути включено до множини B'. Тоді, у загальному випадку, можемо розглядати такі порядки їх обходу, як:

- по мірі збільшення показника $P(\hat{\beta}(ch)_{x,y})$ у діапазоні $\Delta P = \overline{P_{max}; P_{min}}$;

- по мірі зменшення зазначеного показника;

- модифікований напрямок, який може бути реалізовано як та чи інша варіація двох перших напрямків).

У будь-якому випадку, черговість обходу блоків $\hat{\beta}(ch)_{x,y}$ множини B' потребує додаткового внесення у ключ полів, що визначатимуть;

- спосіб обходу;

- параметри обраного способу.

4. РОЗРОБКА ПІДХОДІВ ДО ВНЕСЕННЯ ЦВЗ НА ВИПАДОК МАРКУВАННЯ ВІДЕОКОНТЕНТУ

4.1 Базові вимоги до реалізації способу внесення прихованих міток ЦВЗ у відео потік виходячи зі специфіки його природи

Для захисту графічних даних на базі використання прихованих міток ЦВЗ, у загальному випадку достатньо одноразово виконати їх цифрове маркування.

Разом з тим, існує ряд ключових відмінностей умов розміщення ЦВЗ для випадків графічних даних та відеоконтенту.

У свою чергу, для відеоконтенту, як специфічного типу даних, ткий підхід не є прийнятним з наступних причин [19]:

- відеодані являють собою структуру, що змінюється з часом;
- наближено відеопотік може розглядатися, як сукупність статичних кадрів, що змінюються з плином часу.

З вищезазначеного можемо зробити висновок про те, що:

- неавторизоване використання відеоконтенту може спостерігатися як для тих чи інших відео файлів, так і для їх окремих частин;
- захисту потребують усі ділянки відеопотоку.

При цьому, основними вимогами до способу розміщення ЦВЗ у межах відео можуть бути наступні:

- рівномірний характер внесення ЦВЗ;
- мінімізація навантаження на систему у ході процедури розміщення ЦВЗ.

Для забезпечення можливості розробки підходів до внесення ЦВЗ у відео контент з урахуванням зазначених вимог розглянемо спочатку структуру потоку відеокадрів.

4.2 Загальний огляд типової структура відео потоку

Виходячи з того, що понад 95% відеконтенту сьогодні створюється на засадах MPEG, доцільно брати за основу саме цю концепцію побудови відео потоку.

4.2.1 Головні параметри відеопотоку MPEG

Частіше за все першочерговими параметрами потоку MPEG розглядаються [17, 19]:

- бітова швидкість відео;
- частота fps слідування кадрів за секунду;
- розмірність gop групи кадрів.

Розглянемо кожен з показників більш детально.

Перший з показників – бітова швидкість h , як обсяг біт, що надсилаються за одиницю часу, у нашому випадку не є вирішальним, оскільки обробка виконується на ріні джерела даних.

Далі, якщо говорити про розмірність gop групи кадрів, зазначимо, що у межах потоку MPEG передбачено існування кадрів 3 типів, а саме:

- опорних або базових кадрів I-кадрів (I- intra кадрів);
- двоспрямовано-передбачених, або B-кадрів (bi-predicted);
- передбачених кадрів, або P-кадрів (predicted).

Незалежно від конкретного стандарту у межах MPEG, усі 3 типи присутні у потоці.

Розглянемо для прикладу спочатку потік відеокадрів, що відповідає, т.з. «класичному» Mpeg – технології Mpeg-2 (рис. 4.1).

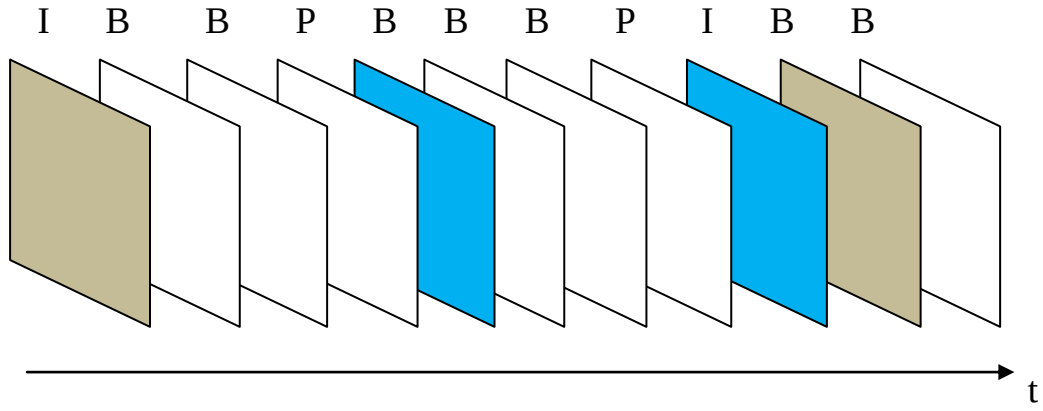


Рисунок 4.1 – Структура потоку кадрів на прикладі однієї групи MPEG-2

Я видно з аналізу рис. 4.1, группа має розмірність 8.

При цьому, основою кожної з груп є кадр I-типу.

Такий кадр кодується повністю, тоді як B і P кадри містять у собі у різних пропорціях інформацію про зміну відео сцени.

Інакше кажучи, кадри B та P типу слугують для передавання різниці між сусідніми I_{ς} та $I_{\varsigma+1}$ відеокадрами (сусідніми базовими кадрами).

Водночас, для рекомендацій H.264/AVC та H.265/HEVC передбачена можливість зміни розмірності величини gop. А саме, $\mathrm{gop}=\overline{8;32}$.

При цьому, слід брати до уваги те, що:

- кадри B-типу частіше за все містять виключно вектори руху, що складає 5-20% від I-кадру [19];

- 99,99% додатків, що надають т.з. «розширений інструментарій» для роботи з відео, здатні зчитувати виключно I-кадри.

При цьому, кадри P-типу є проміжною ланкою між кадрами I та B-типів. У той же час, очевидним є те, що:

- розміщення даних ЦВЗ у межах P та B кадрів є обмежене відносно кадрів I-типу як з точки зору ємності, так і гарантованої цілісності даних;

- у випадку розміщення ЦВЗ виключно у межах P та B кадрів, реалізація процесу їх зчитування перетворюється на завдання, яке потребує залучення додаткових обчислювальних ресурсів (на рівні або приймача, або контролюючої структури).

Разом з тим, на випадок захисту відеоконтенту на базі ЦВЗ рівень захищеності може бути суттєво вищим, ніж для випадку статичних об’єктів. Це зумовлено тим, що:

- у разі розгляду відео як сукупності окремих статичних кадрів, забезпечується можливість обробки лише деякої частини з них, тим самим зменшуючи обчислювальне навантаження, що спричинює на систему алгоритм внесення ЦВЗ;

- у рамках одного кадру може бути внесено лише фрагмент ЦВЗ, що зменшує ємність контейнеру (тим самим збільшуючи його захищеність).

При цьому, з урахуванням особливостей формування кадрів усіх 3 типів, необхідно, у першу, чергу, забезпечити високий рівень робастності алгоритму.

Отже, у загальному випадку, інкапсуляцію ЦВЗ для маркування потоку кадрів відео доцільно виконувати на рівні опорних кадрів.

У свою чергу, частота fps слідування кадрів, у загальному випадку, напряму впливає на показник h бітової швидкості.

Стосовно нашого випадку, урахування даного показника є важливим тому, що за різних комбінацій значень fps та gop кількість опорних кадрів, які попередньо нами обрані для розміщення міток, може суттєво варіюватися.

Далі виконаємо оцінку можливих діапазон кількості ν_I кадрів I-типу у потоці за різних умов в одиницю часу.

4.3 Визначення потенційної кількості опорних кадрів відеопотоку, що підлягають інкапсуляції, за одиницю часу

Зв’язок між розглянутими раніше показниками gop та fps ілюструє наступна схема, як показано рис. 4.2 [19].

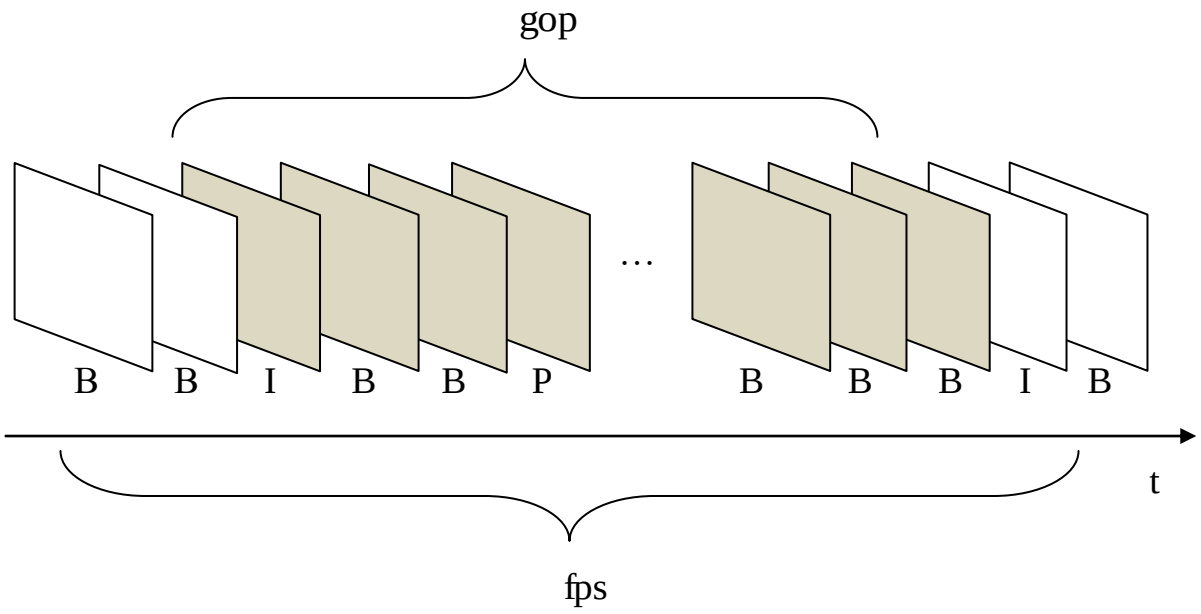


Рисунок 4.2 – Взаємозв’язок між параметрами gop та fps

Як видно з аналізу рис.4.2, в одиницю часу у загальному випадку може бути передано як ряд груп кадрів, так і не повна група.

Зазначимо, що переважна більшість відео контенту, що створюється зараз, кодується на базі H.264/AVC. Тобто, показник gop за даних умов може варіюватися у раніше зазначеному діапазоні. Величина fps для конкретного відеопотоку є сталою, та встановлюється або чинним стандартом, або умовами надання сервісу. Отже, з рис.4.1 та 4.2 можна зробити висновок, що за умови фіксованої величини fps при можливості існування різних значень gop для різних відео файлів, в умовний час спостереження (наприклад, 1 секунду) довільний відрізок відео потоку міститиме у собі різну кількість I-кадрів.

При цьому, незалежно від певної технології у рамках сімейства MPEG, на базі якої сформовано відео потік, або налаштувань у рамках однієї з них, у

межах секундного відрізку часу міститиметься ν_g повних груп кадрів, про що свідчить наступний вираз:

$$\nu_g = \mathrm{div}\left(\frac{\mathrm{fps}}{\mathrm{gop}}\right). \tag{4.1}$$

Водночас, обсяг ν'_g кадрів, що формуватимуть неповну групу, розраховуватиметься згідно з виразом:

$$\nu'_g = \mathrm{fps} - \mathrm{mod}\left(\frac{\mathrm{fps}}{\mathrm{gop}}\right). \tag{4.2}$$

На випадок, коли відео потік кодовано на базі стандарту MPEG-2, маємо $\mathrm{gop} = 8$. При цьому, у випадку, що величина частоти слідування кадрів встановлена у значення $\mathrm{fps} = 8$, відповідно отримуємо $\nu_g = \mathrm{div}\left(\frac{30}{8}\right) = 3$ та $\nu'_g = 30 - \mathrm{mod}\left(\frac{30}{8}\right) = 6$.

При цьому, будь-яка група кадрів починається з кадру I-типу. Тобто, навіть за умови, що $\nu'_g = 1$, неповна група міститиме I-кадр. Тобто, за 1 секунду при $\mathrm{fps} = 8$, у будь якому випадку надсилатиметься:

$$\nu_I = \nu_g + \nu'_g, \tag{4.3}$$

кадрів I-типу. Отже, у даному випадку маємо $\nu_I = 3 + 1 = 4$ опорних кадри.

Разом з тим, при $\mathrm{fps} = 32$ маємо $\nu_g = \mathrm{div}\left(\frac{30}{32}\right) = 0$, у свою чергу, $\nu'_g = 30 - \mathrm{mod}\left(\frac{30}{32}\right) = 0$. Водночас, хоча жодної повної групи за умов, що $\mathrm{fps} = 32$, до секундного інтервалу не буде входити, у будь-якому випадку такий інтервал міститиме 1 кадр I-типу, оскільки група починається саме з нього (рис.3.1).

Таким чином, як попередньо зазначалося, залежно від різних значень величин fps та gop, кількість опорних кадрів за одиницю часу, які потенційно можуть підлягати модифікації у ході внесення ЦВЗ, варіюватиметься.

Водночас, це дає можливість розглянути щонайменше 3 підходи до реалізації алгоритму, а саме:

1. Розміщення ЦВЗ (чи його фрагменту) у кожному з опорних кадрів потоку.

2. Внесення ЦВЗ до одного з I-кадрів у межах елементарного часового інтервалу.

3. Зміна способу вибору кадрів для інкапсуляції за певним механізмом, чи, навпаки, зміна деяким чином у часі з огляду на вплив ряду чинників, зокрема такі, як:

- фактори часу (час доби, коли виконується розміщення ЦВЗ, парність години доби тощо);
- певні параметри потоку;
- середній обсяг окремого контейнеру та ін.

Розглянемо один зі способів вибору кадрів I-типу для внесення ЦВЗ.

4.4 Спосіб вибору кадрів I-типу для внесення ЦВЗ виходячи з параметрів потоку

Нехай існує деякий MPEG-потік (у вигляді файлу чи транслюється), що підлягає захисту. Припустимо, що критерієм захищеності потоку вважається присутність у 3-секундному інтервалі відео щонайменше 2 маркованих кадрів.

При цьому, за умови, що $\mathrm{fps} = 30$, загальна кількість кадрів у зазначеному часовому відрізку для випадку MPEG-2 дорівнюватиме 90, з яких відповідно до виразів (4.1)-(4.3) кількість опорних $\nu_I = 12$.

У свою чергу, для відео, кодованого на базі H.264, цей показник сягатиме 3, ураховуючи повні та одну неповну групу.

Тобто, першим кроком роботи алгоритму вибору кадрів має бути оцінка потенційної величини ν_I.

Далі необхідно визначити, які саме кадри з даної множини підлягають вибору. Тут може бути використано пороговий підхід, сутність якого може бути проілюстрована наступним виразом:

$$\begin{cases} \nu_I^{(1)} \mid \Gamma(r(F_P^m)) \in [\Gamma(r(F_P^m))_{max}; \dfrac{\Gamma(r(F_P^m))_{max}}{\gamma(\Gamma(r(F_P^m)) - 1)}]; \\ \nu_I^{(2)} \mid \Gamma(r(F_P^m)) \in (\dfrac{\Gamma(r(F_P^m))_{max}}{\gamma(\Gamma(r(F_P^m)) - 1)}; \dfrac{\Gamma(r(F_P^m))_{max}}{\gamma(\Gamma(r(F_P^m)) - 2)}]; \\ \ldots \\ \nu_I^{(m)} \mid \Gamma(r(F_P^m)) \in (\dfrac{\Gamma(r(F_P^m))_{max}}{\gamma(\Gamma(r(F_P^m)) - m)}; 0] \end{cases} \quad (4.4)$$

де $\Gamma(r(F_P^m))$ - хеш-функція, ортимана на базі оцінки сукупного рівня $r(F_P^m)$ бітової швидкості від m кадрів F_P^m типу P у межах послідовності, що розглядається;

$\Gamma(r(F_P^m))_{max}$ - деякий максимальний рівень значення хеш-функції у заданих умовах; може встановлюватися довільно, або, наприклад, моделюючи умови, коли усі m кадрів P-типу будуть розглядатися з нульовою квантизацією;

γ - деякий параметричний множник.

Тобто, індекс $\nu_I^{(*)}$ опорного кадру визначається залежно від того, до якого з інтервалів належатиме величина розрахованої хеш-функції.

Такий підхід дозволяє, маніпулюючи величинами γ, m та $\Gamma(r(F_P^m))_{max}$, забезпечити гнучке вбудовування цифрових міток у кадри. Це, у свою чергу, суттєво зменшить ймовірність викриття міток зловмисником. Водночас, уповноваженим особам, які мають відомості щодо γ, m та $\Gamma(r(F_P^m))_{max}$, не буде завдано жодних перешкод для доступу до інформації, що закладена у мітках.

ВИСНОВКИ

У відповідності з технічним завданням, під час виконання кваліфікаційної роботи було виконано:

1.Загальний огляд базових класів даних, що підлягають захисту. При цьому виявлено, що, зокрема, для ПЗ передбачено ряд ефективних механізмів для протидії клонуванню та несанкціонованому використанню і модифікації. Водночас, для мультимедійної інформації такі підходи не можуть бути застосовані.

2. Дослідження базових підходів щодо застосування стеганографічних методів для внесення цифрових міток. Виявлено, що у ряді випадків цифрові мітки застосовуються не як елемент захисту, а як спосіб розміщення даних, супутніх безпосередньо мультимедійному контенту.

3. Виконано дослідження традиційних підходів до розміщення цифрових міток у межах jpeg-контейнерів на прикладі таких стеганографічних методів, як:

- метод на базі використання значень величин компонент трансформованих блоків та його удосконалена модифікація (т.з. метод Langelaar);

- метод модифікації найменш значущого біта (LSB-метод).

У наслідок даного дослідження виявлено, що:

- усі розглянуті методи орієнтуються на використання фронтального методу вбудовування даних;

- особливості контейнеру жодним з методів у повній мірі не ураховуються.

Зазначені недоліки пояснюють низьку стійкість таких методів та їх обмеженість у використанні.

4. Досліджено підхід, у рамках якого реалізація LSB-методу передбачає використання контентно-орієнтованого способу, який дозволяє ураховувати особливості змісту кожної окремої структурної одиниці контейнеру. При цьому, застосовано такі механізми, як:

- механізм вибору структурних одиниць контейнеру з огляду на їхні статистичні характеристики;

- механізм, спрямований на уникнення випадків появи ознак модифікації, властивих для випадку фронтального вбудовування.

5. Розроблено концепцію підходу, що орієнтований на захист відео контенту на базі внесення цифрових міток в опорні кадри.

Отже, усі пункти технічного завдання у виконано у мовній мірі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Конахович Г. Ф., Пузыренко А. Ю. Компьютерная стеганография. Теория и практика. - К.: МК-Пресс, 2006. - 288 с
2. Шаньгин В.Ф. Информационная безопасность и защита информации. ДМК-Пресс., 2017, 702 с.
3. Шелухин О.И., Канаев С.Д. Стеганография. Алгоритмы и программная реализация. Горячая линия – Телеком, научно-техническое издательство 2017, 592 с.
4. Рябко, Б.Я. Основы современной криптографии и стеганографии [Электронный ресурс] : [монография] / А.Н. Фионов, Б.Я. Рябко. — М. : Горячая линия – Телеком, 2010 .— 233 с.
5. Грибунин В. Г., Оков И. Н., Туринцев И. В. Цифровая стеганография. М.: СОЛОН-Пресс, 2016, - 315 с.
6. Быков С. Ф. Алгоритм сжатия JPEG с позиции компьютерной стеганографии Защита информации. Конфидент. - СПб.: 2000, № 3
7. Цифровая стеганография: Программы и другие способы реализации [Електронний ресурс] – Режим доступу: http://www.spy-soft.net/cifrovaya-steganografiya-sposoby-realizacii/
8. GitHub - x2on/FSImageViewer: Photo viewer (gallery) for iOS with AFNetworking and caching [Електронний ресурс] – Режим доступу: https://github.com/x2on/FSImageViewer
9. Camouflage Home Page - Hide your files! [Електронний ресурс] – Режим доступу: http://camouflage.unfiction.com/
10. Алексеев, А.П. Стеганографические и криптографические методы защиты информации : учеб. пособие по дисциплине "Информатика" / В.В. Орлов, А.П. Алексеев .— Самара : ИУНЛ ПГУТИ, 2010 .— 289 с.
11. Юренский П.В. МЕТОДЫ СТАТИСТИЧЕСКОГО И НЕЙРОСЕТЕВОГО СТЕГОАНАЛИЗА СКРЫТЫХ КАНАЛОВ // Инновации в науке: научный журнал. – № 1(89). – Новосибирск., Изд. АНС «СибАК», 2019. – С. 11-13.
12. DiffImg download | SourceForge.net [Електронний ресурс] – Режим доступу:https://sourceforge.net/projects/diffimg/
13. Dumitrescu, S., W. Xiaolin and Z. Wang, 2003. Detection of LSB steganography via sample pair analysis. In: LNCS, Vol. 2578, Springer-Verlag, New York, pp: 355-372.

14. Fridrich Y. Steganography in Digital Media: Principles, Algorithms and Applicaticks. Cambridge Press, 2010. 462 p
15. Shi, Yun Q. Image and video compression for multimedia engineering: fundamentals, algorithms, and standards / Yun Q Shi, Huifang Sun
16. Ватолин Д. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М. : ДИАЛОГ – МИФИ, 2003. – 384 с
17. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов / В. Г. Олифер, Н. А. Олифер. – 3-е изд. – СПб. : Питер, 2006. – 958 с.
18. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1073 с.
19. Айфичер Эммануил С. Цифровая обработка сигналов: практический поход / Эммануил С. Айфичер, Барри У. Джервис. – 2-е изд. – М. : Вильямс, 2008. – 992 с.
20. Чемпен Н., Чемпен Д. Цифровые технологии мультимедиа. – М.: Вильямс, 2006. – 624 с.
21. Красильников Н.Н. Цифровая обработка изображений. – М.: Вузовская книга, 2011. – 320 с.
22. Сэломон Д. Сжатие данных, изображений и звука / Д. Сэломон. – М.: Техносфера, 2004. – 368 с.
23. Ричардсон Ян. H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia / Ян Ричардсон. – Город. : Издательство, 2005. – 368 с.
24. Баранник В.В. Кодирование трансформированных изображений в инфокоммуникационных системах / В.В. Баранник, В.П. Поляков – Х.: ХУПС, 2010. – 234 с.
25. Красильников Н.Н. Цифровая обработка изображений. – М.: Вузовская книга, 2011. – 320 с.