Міністерства освіти і науки україни
Одеський національний політехнічний університет
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Чіклікчі Іван Сергійович
студент групи РЗ-151

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Розробка методу контролю цілісності зображення на основі цифрового водяного знака

Спеціальність:
125 Кібербезпека

Спеціалізація, освітня програма:
Кібербезпека

Керівник:
Кушніренко Наталія Ігорівна
доцент

Одеса – 2020

АНОТАЦІЯ

Чіклікчі І.С. Розробка методу контролю цілісності зображення на основі цифрового водяного знака – Магістерська кваліфікаційна робота. Одеса, 2020: 72 с., 12 рис., 1 табл., 16 джерел.

Об'єкт дослідження – системи контролю цілісності зображень.

Предмет дослідження – Метод контролю цілісності зображень, що ґрунтуються на використанні цифрових водяних знаків.

Мета роботи – є розробка стеганографічних метода перевірки цілісності зображення, який базується на хеш функції від зображення, методам можна впровадити в зображення результат хеш функцій який гарантуватиме що над зображення не застосовували різного роду операцій.

1. Провести аналіз моделей і методів контролю цілісності зображень, заснованих на використанні цих моделей і методів;
2. Дослідити можливості зменшення кількості змінюваних бітів зображення;
3. Запропонувати новий метод контролю цілісності зображень з вбудовування хеш значення;
4. Розробити програмне забезпечення, яке реалізує запропонований метод контролю цілісності;
5. Експериментальним шляхом дослідити кількість змінюваних бітів в зображенні при встановленні хеш значення.

КОНТРОЛЬ ЦІЛІСНОСТІ РАСТРОВІХ ЗОБРАЖЕНЬ, ЦИФРОВИЙ ВОДЯНОЙ ЗНАК, СТЕГОГРАФІЧНИЙ КЛЮЧ, ВБУДОВУВАННЯ СЕКРЕТНОЇ ІНФОРМАЦІЇ, ХЕШ-СУМА.

ANNOTATION

Chiklikchi I.S. Development of a method for controlling the integrity of the image based on a digital watermark - Master thesis. Odessa, 2020: 72 pp., 12 fig., 1 tab., 16 sources.

Object of study - image integrity control systems.

Subject of research - method for controlling the integrity of raster images based on the use of digital watermarks.

The purpose of the work - is to develop steganography methods for checking the integrity of the image, which is based on the hash function from the image, the methods can be implemented in the image result of hash functions, which will ensure that the image is not applied to various operations.

1. Analyze models and methods of image integrity control based on the use of these models and methods;

2. Investigate the possibility of reducing the number of variable bits of the image;

3. To propose a new method of image integrity control by embedding hash values;

4. Develop software that implements the proposed method of integrity control;

5. Experimentally investigate the number of variable bits in the image when setting the hash value.

CONTROL OF INTEGRITY OF RASTER IMAGES, DIGITAL WATER SIGN, STEGO KEY, INTEGRATION OF SECRET INFORMATION, HASH AMOUNT.

ЗМІСТ

ВСТУП

Актуальність. Питання перевірки достовірності предметів, таких як гроші, картини, вироби і т.д, все більше стає актуальним з розвитком інформаційних технологій. В даний час у зв'язку з великим розвитком мережевих технологій автоматична аутентифікація використовується повсюдно та хеш-функції - є гарантом надійності, тому що їх значення є незворотнім для кожного зображення.

При впровадженні цифрових водяних знаків (ЦВЗ) в зображення деякі його елементи неминуче зазнають зміни. Таким чином, в існуючих системах ЦВЗ йдеться про достовірність встановлення джерела поширення зображення, а не про його інформативної цілісності, яка порушується вже в момент додавання в зображення будь-якого ідентифікаторів. Однак, для деяких видів зображень гарантія цілісності є одним з пріоритетних критеріїв системи ЦВЗ. Цілісність зображення в такому випадку може гарантуватися так само, як і для інших типів файлів - шляхом хеширования файлу зображення.Таким чином, розробка нового методу основого на хеш технології є надзвичайно важливою і актуальним завданням.

Мета і задачі досліджень. Метою роботи є розробка стеганографічних метода перевірки цілісності зображення, який базується на хеш функції від зображення, методам можна впровадити в зображення результат хеш функцій який гарантуватиме що над зображення не застосовували різного роду операцій.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

6. Провести аналіз моделей і методів контролю цілісності зображень, заснованих на використанні цих моделей і методів;
7. Дослідити можливості зменшення кількості змінюваних бітів зображення;
8. Запропонувати новий метод контролю цілісності зображень з вбудовування хеш значення;
9. Розробити програмне забезпечення, яке реалізує запропонований метод контролю цілісності;

10. Експериментальним шляхом дослідити кількість змінюваних бітів в зображенні при встановленні хеш значення.

Об'єкт дослідження – системи контролю цілісності зображень.

Предмет дослідження – методи контролю цілісності зображень, що ґрунтуються на використанні цифрових водяних знаків.

Практичні значення методу. Розробка у кваліфікаційній роботі у подальшому можуть бути використані для для верифікації зображень, джерела зображення, цілісності зображення.

Публікації. Основні положення та результати поданої роботи представлені у журналі «Інформатика та математичні методи в моделюванні», 2020.

1. АНАЛІТИЧНИЙ ОГЛЯД МЕТОДІВ КОНТРОЛЮ ЦІЛІСНОСТІ ЗОБРАЖЕНЬ

1.1. Опис контролю цілісності

В основі контролю цілісності лежать два поняття:

- хеш-функція;
- електронний цифровий підпис (ЕЦП).

Криптографічна геш-функція — це геш-функція, яка є алгоритмом, що приймає довільний блок даних і повертає рядок встановленого розміру, значення якої залежить від принятого блоку, при зміни даних змінять геш-значення.[1].

Нехай є дані, цілісність яких потрібно перевірити, хеш-функція і раніше обчислений результат її застосування до вихідних даних (так званий дайджест). Позначимо хеш-функцію через *Y*, вихідні дані - через *V*, перевіряються дані - через *V'*. Контроль цілісності даних зводиться до перевірки рівності *Y(V') = V (V).* Якщо воно виконано, вважається, що *V '= V* [1].

Збіг дайджестів для різних даних називається колізією. В принципі, колізії, звичайно, можливі, оскільки потужність множини дайджестів менше, ніж потужність безлічі хешируемих даних, однак те, що *Y* є функція одностороння, означає, що за прийнятний час спеціально організувати колізію неможливо.

Електронний цифровий підпис захищає цілісність повідомлення та засвідчує особу відправника, тобто захищає цілісність джерела даних і служить основою недовідності.

Використання асиметричного шифрування має істотний недолік. Зловмисник може, видаючи себе за іншого користувача, сформувати пару ключів (відкритий і закритий), опублікувати відкритий ключ і отримувати повідомлення, адресовані цьому користувачу. Для усунення цього недоліку застосовуються цифрові сертифікати.

Для вироблення і перевірки ЕЦП необхідно виконання наступної умови:

$$E'(D(V)) = D'(E(V)) = V \tag{1.1}$$

де *V* - зашифроване повідомлення,

D - результат шифрування *V* секретним ключем,

E - результат шифрування *V* відкритим ключем,

D' - результат дешифрування *V* за допомогою секретного ключа,

E' - результат дешифрування *V* за допомогою відкритого ключа.

Цифровиий сертифікат — цифровий документ, який є одним із засобів підтвердження відкритого ключа приналежності його власникові. Найчастіше електронний сертифікат додається до підписаного ЕЦП електронного документа та використовується для перевірки, що відкритий ключ належить саме тому власнику, чиї дані зазначено в ньому.

Користувач деяким захищеним чином доставляє в ЦС свій відкритий ключ і підтверджує свою справжність. ЦС формує сертифікат. Отримавши сертифікат користувач публікує його. Якщо відкритий ключ користувача потрібен іншому користувачеві для передачі повідомлень, то він може переконатися в достовірності сертифікату, перевіривши ЕЦП. [1].

Сертифікат містить:

- серійний номер сертифіката;
- назва алгоритму цифрового підпису;
- назва центра сертифікації, що підтвердив підпис власника;
- термін дії сертифікату з та по;
- ім'я користувача, якому належить сертифікат;
- відкритий (публічний) ключ власника сертифіката (ключів може бути декілька);
- об'єктні ідентифікатори алгоритмів, асоційованих з відкритими ключами власника сертифіката;
- електронний цифровий підпис центру сертифікації вищенаведених даних. Всі дані, які й власно утворюють сертифікат, отримують цифровій відбиток, якій відбивається в сертифікаті;

– назву видавця (issuer).

Для формування ЕЦП виконуються наступні дії:

– виконується хеш-перетворення повідомлення *V - Y (V)*;

– результат перетворення шифрується секретним ключем ЦС - *D (Y (V))*.

Для перевірки ЕЦП:

– за допомогою відкритого ключа дешифрується підпис *V'(D(Y (V)))=Y(V);*

– виконується хеш-перетворення повідомлення *V'- Y (V');*

– перевіряється рівність *Y (V) = Y (V')*.

Засоби контролю цілісності даних застосовуються для контролю інформаційних ресурсів, файлів, каталогів і папок, при якому забезпечується їх незмінність, захист від модифікації і спотворення, протягом усього життєвого циклу. Цей параметр є критичним аспектом проектування, реалізації та використання будь-якої інформаційної системи, яка зберігає, обробляє або отримує дані. Головна мета забезпечення цілісності даних полягає в тому, щоб інформація змінювалася тільки запланованим чином, а також зберігалася незмінною при наступному зверненні, або, іншими словами, була захищена від несанкціонованих дій. Будь-які ненавмисні зміни даних, що виникають в результаті операцій зберігання, вилучення чи обробки, включаючи впливу шкідливих програм, несподівана відмова обладнання або людський фактор - призводять до помилок цілісності даних. В інформаційних системах цілісність даних, які не потрапляють за межі системи, забезпечується за рахунок відмово стійкості і можливості відновлення даних, наприклад, з резервних копій. У разі відправлення файлів в зовнішні інформаційні системи можуть застосовуватися криптографічні методи захисту інформації.

Забезпечення цілісності даних також є однією з функцій засобів захисту інформації від несанкціонованого доступу. Ці засоби захисту виробляють відстеження незмінності даних в автоматичному режимі за заздалегідь налаштованому розкладом. При цьому крім незмінності самих файлів, засоби захисту можуть перевіряти незмінність прав доступу до об'єктів і їх атрибутів. У разі виявлення помилок при проходженні процедури контролю цілісності, засоби

захисту від несанкціонованого доступу можуть сигналізувати про це адміністраторам безпеки, забороняти доступ до системи, зберігати зміни в файлах або відкочувати ресурси до початкового стану. При цьому засоби захисту від несанкціонованого доступу надають такі алгоритми перевірки цілісності даних: електронно-цифровий підпис (перевіряється вбудована цифровий підпис даних), CRC32 (або розрахунок контрольних сум файлів), хеш, имитовставка, повний збіг.

Цілісність інформації - термін в інформатиці (криптографії, теорії телекомунікацій, теорії інформаційної безпеки), що означає, що дані не були змінені під час будь-якої операції над ними, будь то передача, зберігання або відтворення. В телекомунікації цілісність даних часто перевіряють, використовуючи хеш-суму повідомлення, обчислену алгоритмом MAC (англ. Message authentication code). У криптографії та інформаційної безпеки цілісність даних (в широкому сенсі) - це збереження даних в тому вигляді, в якому вони були створені. Приклади порушень цілісності даних:

- спроба зловмисника змінити номер аккаунта в банківській транзакції, або спроба підробки документа;
- випадкове зміна інформації при передачі або при несправної роботі жорсткого диска;
- перекручування фактів засобами масової інформації з метою маніпуляції громадською думкою.

В теорії баз даних цілісність даних означає коректність даних і їх несуперечливість. Зазвичай вона також включає цілісність зв'язків, яка виключає помилки зв'язків між первинним і вторинним ключем. Приклади порушень цілісності даних:

- існування записів-сиріт (дочірніх записів, які не мають зв'язку з батьківськими записами);
- існування однакових первинних ключів.

Для перевірки цілісності даних в криптографії використовуються хеш-функції, наприклад, SHA-1. Хеш-функція перетворює сукупність електронних даних довільного розміру в електронних даних фіксованого розміру (число).

Якщо дані зміняться, то і число, що генерується хеш-функцією, теж зміниться. Цілісність даних - властивість, при виконанні якого дані зберігають заздалегідь певний вид і якість. При цьому крім незмінності самих файлів, засоби захисту можуть перевіряти незмінність прав доступу до об'єктів і їх атрибутів. У разі виявлення помилок при проходженні процедури контролю цілісності, засоби захисту від несанкціонованого доступу можуть сигналізувати про це адміністраторам безпеки, забороняти доступ до системи, зберігати зміни в файлах або відкочувати ресурси до початкового стану.

Методи і способи реалізації вимог, викладених у визначеннях терміна, докладно описуються в рамках єдиної схеми забезпечення інформаційної безпеки об'єкта (захисту інформації). Основними методами забезпечення цілісності інформації (даних) при зберіганні в автоматизованих системах є:

- забезпечення відмовостійкості (резервування, дублювання, віддзеркалення обладнання та даних, наприклад через використання RAID-масивів);
- забезпечення безпечного відновлення (резервне копіювання і електронне архівування інформації).

Одним з дієвих методів реалізації вимог цілісності інформації при її передачі по лініях зв'язку є криптографічний захист інформації (шифрування, хешування, електронний цифровий підпис). При комплексному підході до захисту бізнесу, напрям забезпечення цілісності та доступності інформації (ресурсів бізнес-процесів) переростає в план заходів, спрямованих на забезпечення безперервності бізнесу [2].

Для контролю цілісності частіше використовують чотири метода. Три метода дуже легкі у реалізації проте не є надійними, четвертий метод основа на використані цифрового водяного знаку, що робить цього більш надійним але він більш складний у реалізації.

Перший метод контролю цілісності растрових зображень базується на зберіганні хеш-суми поруч з картинкою, при необхідності перевірки оригінальності зображення, іхображеніе знову пропускають через хеш-функцію і

звіряють нову хеш-суму з уже раніше збереження якщо вони збігаються тоді можна затверджувати що картинка не була піддана змінам. Але так як хеш-сума зберігається поряд з зображення стає очевидним що зображення подрегнуто контролю цілісності, в зв'язку з цим злоумишлінікам не має відбутися праці змінити картинку пропустити вже відредаговану фотографію через хеш-функцію і перезаписати раніше збереження хеш-суму тоді при перевірці достовірності зображення результат буде помилково позитивним. У зв'язку з цим недоліком даний метод не варто використовувати, так як він не надійний. Гідність методу в його легкої реалізації.

Другий спосіб контролю цілісності зображення зберігання хеш суми у віддаленій базі даних, при необхідності перевірки картинки на цілісність картинка знову пропускається через хеш функцію і звіряється з уже збереження в базі даних, в даному випадку злоумишлінік не знає про те що картинка контролюється це вважається плюсом. Але можна отримати доступ до віддаленої бази і підмінити хеш суму зміни картинки в базі або підключиться до запиту з бази даних і зраджувати потрібну хеш суму щоб результат перевірки був помилково позитивним і вважали що картіннка оригінал. Зручність у даному методі то що зловмисник не знає про контроль картинки але все ж хеш суму можна отримати з бази що ні надійно.

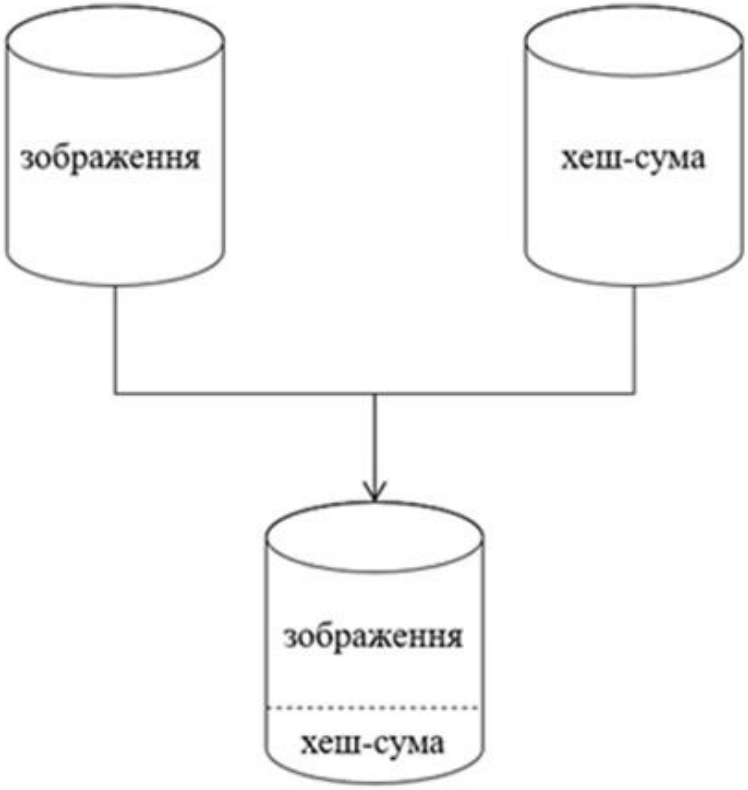

Рисунок 1.2. - Зєднання хеш-суми і зображення

Третій спосіб контролю цілісності зображення базується на використанні хеш-суми яка поєднується із зображенням в один файл. хеш-сума в даному випадку приклеюватися до картинки, в дано випадку легко дістати хеш функцію цю хеш функию не видно і не зрозуміло дана картинка подліжіт контролю цілісності чи ні, але так само легко дістати її змінити і вписати нову хеш функцію що не є надійним.

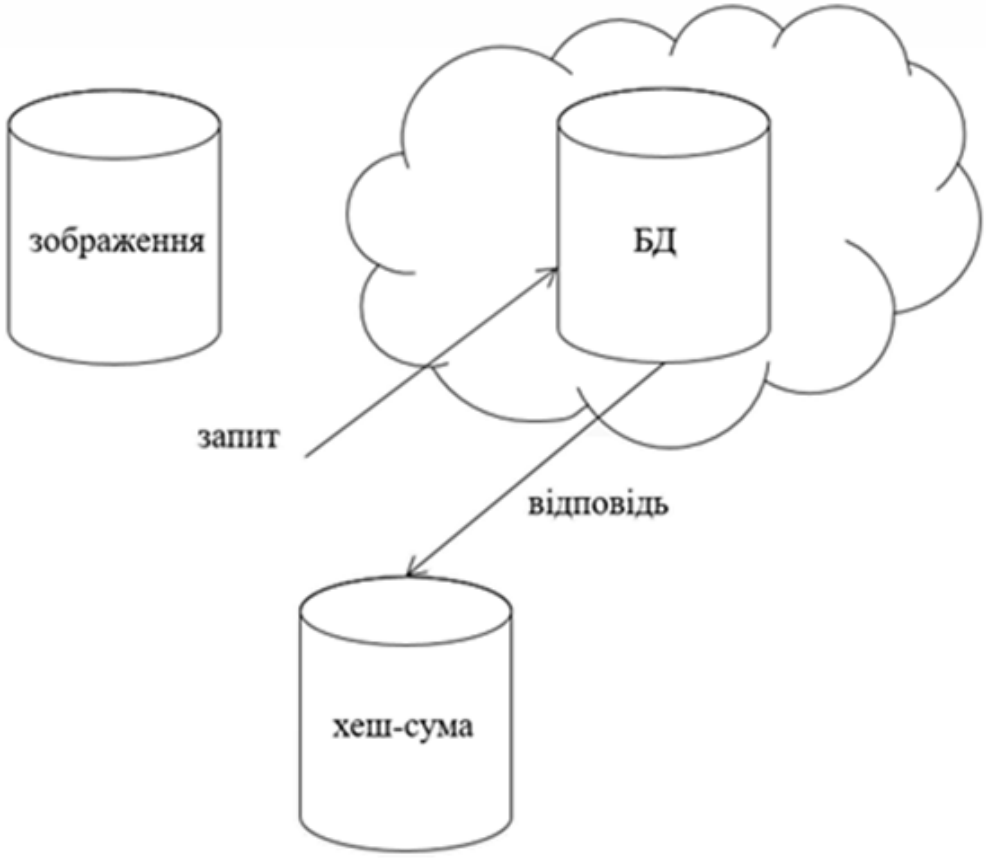

Рисунок 1.3. - Зберіганні хеш-суми в базі даних

Четвертий спосіб контролю цілісності використовує стенографічний метод, в даному випадку злоушленік ні яким чином не зможе зрозуміти що картинка піддана контролю цілісності, в даному методі так само використовується хеш сума картинки тільки даний метод пропонує використовувати як сховище картинки саму картинку. Для того щоб заховати хеш суму використовують молодший розряд одного з каналів кольорів. В даному методі перезаписувати молодші розряди пікселів одного з квітів що впливає на картинку так як молодший розряд зберігає збільшує значення кольору пікселя на одне значення в більшу так і в меншу сторону, але людське око не може визначити ці зміни в кольорі так як вони нічтожно малі, навіть якщо звіряти з оргіналом немає можливості їх розрізнити. Складність даного методу ще в тому що потрібно записувати не тільки хеш суму але і потрібно помістити всі використовувані молодші біти, якщо використовувати 256 розрядну хеш суму і припустимо що наша картинка має 1000 пікселів то нам необхідно назад помістити тисяча двісті

п'ятьдесят шість біт що не можливо зробити, для цього використовується стиснення даних без втрат ми стискаємо наші тисячі пікселів для востановления і виходить 500 пікселів що в підсумку нам потрібно записати 756 пікселів.

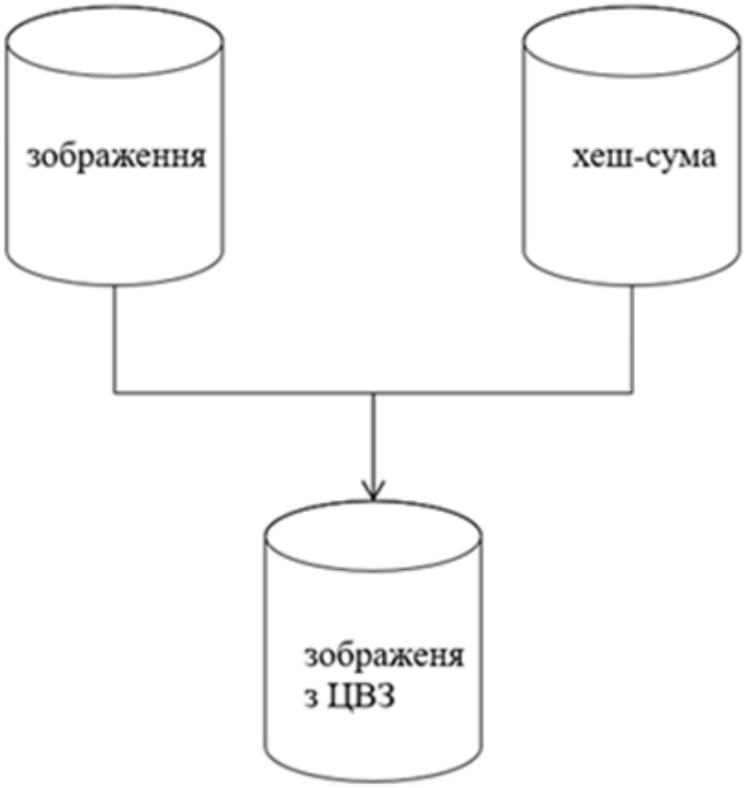

Рисунок 1.3. - Використання цифрового водяного знаку

Не достаток даного методу в тому що не завжди після стиснення ми отримуємо буфер достатнього розміру щоб записати туди нашу хеш суму, це часто виходить з картинками малого дозволу, так само не можливо використовувати даний метод з форматами картинок як jpeg.

1.2. Основні формати цифрових зображень

Під цифровим зображенням слід розуміти надання інформації в графічному вигляді, яке призначене для зорового сприйняття. При цьому цифрове зображення може спочатку створюватися в цифровому вигляді з використанням комп'ютерної

програми або бути перетвореним з природного або аналогового видів в цифровий за допомогою пристроїв введення.

Спосіб запису і зберігання графічної інформації у файлі називається графічним форматом. Формати графічних файлів досить сильно відрізняються один від одного в залежності від типу інформації, що зберігається в них інформації[3].

Всі існуючі цифрові зображення за принципом їх формування, залежному від зберігається в них інформації, можна розділити на чотири види: фрактальна, тривимірна, векторна і растрова графіка. По виду прив'язки до типу зображення графічні формати можна розбити на два види: формати, що представляють спеціалізовані зображення з чіткою структурою, і формати, що не пред'являють ніяких вимог до характеру зображень[3]

Растрове зображення розглядається людським мозком як двовимірна матриця, основним елементом якої є точка або піксель, що характеризується кольором і координатами в горизонтальному і вертикальному рядах зображення. Для запису відповідного кожного пікселя оптичного сигналу використовують різні способи, найбільш поширеним з яких є розкладання сигналу по його спектральним складовим.

Розробники пропонують чимало растрових форматів, призначених для зберігання файлів. Серед найбільш часто використовуваних варто назвати наступні: BMP, TIFF, GIF, JPEG, PNG, PSD, ICO. Отже, розглянемо деякі плюси і мінуси, а також область застосування перелічених растрових форматів зображень:

– BMP є стандартним растровий формат і має універсальне призначення. Він підтримується більшістю графічних редакторів, включаючи досить поширений Paint. Спочатку кодування в ньому виконувалося найпростішим способом, по пікселям. Але це виявилося неекономно, оскільки кожен піксель був представлений лише одним байтом. Отже, ставали доступними всього 256 кольорів, що істотно обмежувало можливості передачі зображень. Надалі він кілька удосконалювався. Bit Map image майже оптимально підходить для зберігання даних і обміну

ними з іншими подібними програмами. Але, разом з тим, займає надто багато місця в пам'яті, так як необхідно зберігати кодування всіх точок зображення;

- TIFF універсальний для видавничих систем і топографічної графіки. Такі формати растрових зображень забезпечують високу якість друку. Вони створювалися для підтримки практично всіх програм, призначених для роботи з файлами точкової графіки, тому поєднуються з усіма платформами. Широко використовують TIFF в поліграфії та видавничій справі. Файли (відскановані зображення, ілюстрації, факси тощо) з розширенням .tif в цьому потужному форматі зберігають для подальшої кольорового друку, хоча доступна і монохромна роздруківка - в уявленнях CMYK і RGB. Чи не застосовується для публікації картинок в комп'ютерній мережі або при створенні веб-сайтів, адже має досить значні розміри. Непридатний він також для анімації;
- GIF служить для зберігання растрових зображень в графіку і для обміну ними. Він один з найбільш «старих» в Інтернеті, має ходіння вже тривалий час, незважаючи на те, що в ньому застосовуються індексовані кольори (в обмеженому наборі). Файли з розширенням .gif широко використовують при конструюванні Web-сайтів. Серед основних плюсів Graphic Interchamge Format варто назвати те, що вид картинки не залежить від базової платформи або від типу браузера, а стиснення відбувається без втрат інформації. Високоякісно в цьому форматі відображаються малюнки з незначною кількістю однорідних кольорів, креслення, прозорі картинки та анімація. Але все ж формат має істотний недолік - у нього незначний набір квітів, що обмежує його можливості при зберіганні зображень, у яких плавні переходи;
- JPEG допомагає позбутися від вад, які виникають при створенні та збереженні зображень в GIF. Тут використовується метод стиснення фотографій або інших картинок. Ці формати растрових графічних файлів є найбільш поширеними при зберіганні багатобарвних картинок.

Стиснення зображень (вони зберігаються в файлах з позначкою .jpg) Виконується в плавному режимі, що забезпечує високу його ступінь і знижує втрати даних. На жорсткому диску в JPEG зручно зберігати значну кількість картинок, зокрема - великі фотознімки з плавними переходами. Це дозволяє істотно заощадити місце на диску. Також за допомогою JPEG є публікувати цілком прийнятної якості фото в комп'ютерній мережі. Але слід враховувати, що при стисненні частина даних втрачається, а при повторному збереженні того ж зображення шанси незворотною втрати інформації зростають. У цьому плані набагато покращує становище вдосконалена версія формату - JPEG 2000. Правда, підтримується він не всіма браузерами, що гальмує його поширення;

- PNG дозволяє зберігати растрові дані в стислому вигляді без втрат, причому файли виходять менше за обсягом, ніж в GIF. У форматі PNG є застосування практично будь-якого кольору, а також прозорість. Ця обставина розкриває широкі можливості в веб-конструюванні. Зараз користується постійною популярністю, оскільки стикується з усіма платформами підтримує черезстрочная відображення, відрізняється значною кольоровою гамою, підтримує анімацію;
- Внутрішні формати растрової графіки PSD (скорочення від PhotoShop Document) призначені для пакетів програми Adobe Photoshop. Вони підтримують всі типи зображень, а також їх шари в ході обробки. Зберігаються в файлах з позначкою розширення .psd;
- Формат файлів ICO (Windows icon) використовується програмами для створення картинок малого розміру (так званих «іконок») в браузерах комп'ютерних систем. Іконками маркуються веб-проекти в рядку «Вибране» або URL;
- Формат файлів RAW часто називають «сирим» графічним форматом і порівнюють з плівковим негативом.

1.3 Розгляд принципів стеганографії

Стеганографія - наука про захист інформації шляхом приховування факту передачі повідомлення.

У сучасній стеганографії існує два основних типи файлів: повідомлення - файл, що призначений для приховування, і контейнер - файл, що може бути використаний для приховування в ньому повідомлення. При цьому контейнери бувають двох типів. Контейнер-оригінал - це контейнер, що не містить схованої інформації. Контейнер-результат - це контейнер, що містить сховану інформацію. Під ключем розуміється секретний елемент, що визначає порядок занесення повідомлення в контейнер. Класичним є наступний принцип вбудовування даних. Нехай сигнал контейнера представлений послідовністю з N біт. Процес приховання інформації починається з визначення біт контейнера, які можна змінювати без внесення помітних спотворень — стеганошляху. Далі серед цих біт, зазвичай у відповідності до ключа, обираються біти, що замінюються бітами приховуваного повідомлення.

Раніше широко використовувалися так звані симпатичні чорнила, невидимі при звичайних умовах. Приховане повідомлення розміщували в певні букви невинних словосполучень, передавали за допомогою внесення в текст незначних стилістичних, орфографічних або пунктуаційних помилок. Крапчасті карт шулерами - це теж приклад стеганографії. Приховування інформації перерахованими методами можливо лише завдяки тому, що противнику невідомий метод приховування. Вся секретність системи захисту переданих відомостей заснована на ключі, тобто на попередньо (як правило) розділеному між адресатами фрагменті інформації. Бо тільки застосовуючи ключ можно дізнатися приховану інформацію. Звичайно, вони не можуть застосовуватися в серйозних цілях. Розвиток засобів обчислювальної техніки дало новий поштовх для розвитку комп'ютерної стеганографії. З'явилося багато нових областей застосування. Повідомлення вбудовують тепер в цифрові дані, як правило, мають аналогову природу такі як мова, аудіозаписи, зображення, відео. Відомі також

пропозиції по врахуванню інформації в текстові файли і в виконувані файли програм.

За способом обрання контейнера розрізняють сурогатні, селективні та імітаційні методи стеганографії. В сурогатних методах стеганографії відсутня можливість вибору контейнера і для приховування повідомлення вибирається перший контейнер, що трапився, який у більшості випадків не є оптимальним для приховуваного повідомлення. У селективних методах стенографії передбачається, що приховане повідомлення повинно відтворювати спеціальні статистичні характеристики шуму контейнера. Для цього генерують велику кількість альтернативних контейнерів, з наступним обранням (шляхом відбраковування) найоптимальнішого з них для конкретного повідомлення. Засоби захисту інформації Вісник Національного технічного університету випадком такого підходу є обчислення деякої хеш-функції для кожного контейнера. При цьому для приховання повідомлення обирається той контейнер, хеш-функція якого збігається зі значенням хеш-функції повідомлення (тобто стеганограмою є обраний контейнер). В імітаційних методах стеганографії контейнер генерується самою стеганосистемою. При цьому існують декілька варіантів реалізації. Так, наприклад, шум контейнера може імітуватися приховуваним повідомленням. Це реалізується за допомогою процедур, які не лише кодують приховане повідомлення під шум, але й зберігають модель початкового шуму. У граничному випадку за моделлю шуму може будуватися ціле повідомлення. Прикладом може слугувати метод, реалізований у програмі MandelSteg, яка в якості контейнера генерує фрактал Мандельброта (Mandelbrot fractal), або ж апарат функцій імітації.

Цифрові водяні знаки можуть застосовуватися, в основному, для захисту від копіювання та несанкціонованого використання. У зв'язку з бурхливим розвитком технологій мультимедіа гостро постало питання контролю цілісності захисту авторських прав та інтелектуальної власності, представленої в цифровому вигляді. Прикладами можуть бути фотографії, аудіо і відеозаписи і т.д. Переваги, які дають уявлення і передача повідомлень в цифровому вигляді, можуть виявитися покарбованими с легкістю, з якою можливо їх злодійство або модифікація. Тому

розробляються різні заходи захисту інформації організаційного і технічного характеру. Одне з найбільш ефективних технічних засобів захисту мультимедійної інформації полягає у вбудовуванні в об'єкт, що захищається невидимих міток - цифрових водяних знаків. Розробки в цій галузі ведуть найбільші фірми в усьому світі.

На відміну від звичайних водяних знаків цифрові водяні знаки можуть бути не тільки видимими, але і (як правило) невидимими. Невидимі цифрові водяні знаки аналізуються спеціальним декодером, який виносить рішення про їх коректності. Цифрові водяні знаки можуть містити деякий автентичний код, інформацію про власника, або яку-небудь інформацію, що управляє. Найбільш придатними об'єктами захисту за допомогою цифрового водяного знаку є нерухомі зображення, файли аудіо- і відеоданих. Технологія вбудовування ідентифікаційних номерів виробників має багато спільного з технологією цифрового водяного знаку. Відмінність полягає в тому, що в першому випадку кожна захищена копія має свій унікальний вбудований номер звідси і назва fingerprinting - дослівно «відбитки пальців». Цей ідентифікаційний номер дозволяє виробникові відстежувати подальшу долю свого дітища: чи не зайнявся хтось із покупців незаконним тиражуванням. Якщо так, то «відбитки пальців» швидко вкажуть на винного.

Стегосистема з цифровим водяним знаком містить наступні ключові вузли:

- прекодер - пристрій, призначений для перетворення приховуваного повідомлення до вигляду, зручного для вбудовування в сигнал-контейнер (інформаційна послідовність, в якій ховається повідомлення);
- стегокодер - пристрій, призначений для здійснення вкладення прихованого повідомлення в інші дані з урахуванням їх моделі;
- пристрій виділення вбудованого повідомлення;
- стегодетектор - пристрій, призначений для визначення наявності стегосообщенія;
- декодер - пристрій, що відновлює приховане повідомлення (даний вузол може бути відсутнім).

Дані, що містять приховане повідомлення, можуть піддаватися навмисним атакам або випадковим перешкод. У стегосистеми відбувається об'єднання двох типів інформації так, щоб вони могли бути помітні двома принципово різними детекторами. В якості одного з детекторів виступає система виділення цифрового водяного знаку, як іншого - людина. Для того, щоб підвищити стійкість цифрового водяного знаку до спотворень нерідко виконують його завадостійке кодування, або застосовують широкосмугові сигнали. Первісну обробку прихованого повідомлення виконує прекодер. В якості найважливішої попередньої обробки цифрового водяного знаку (а також і контейнера) назвемо обчислення його узагальненого перетворення Фур'є. Це дозволяє здійснити вбудовування цифрового водяного знаку в спектральної області, що значно підвищує його стійкість до спотворень. Попередня обробка часто виконується з використанням ключа для підвищення секретності вбудовування. Далі цифровий водяний знак «вкладається» в контейнер, наприклад, шляхом модифікації молодших значущих біт коефіцієнтів. Цей процес можливий завдяки особливостям системи сприйняття людини. Добре відомо, що зображення мають велику псіховізуальной надмірністю. Око людини подібний до низькочастотному фільтр, що пропускає дрібні деталі. Особливо непомітні спотворення в високочастотної області зображень. У більшості стегосістем для впровадження і виділення цифрового водяного знаку використовується ключ. Ключ може бути призначений для вузького кола осіб або ж бути загальнодоступним. Наприклад, ключ повинен міститися у всіх DVD-плеєрах, щоб вони могли прочитати містяться на дисках цифровий водяний знак. Іноді за аналогією з криптографією стегосистеми ділять на два класи: з відкритим ключем і з секретним ключем. У стегодетекторе відбувається виявлення цифрового водяного в (можливо зміненому) захищеному цифровим водяним знаком зображенні. Ця зміна може бути обумовлено впливом помилок в каналі зв'язку, операцій обробки сигналу, навмисних атак порушників. Тоді задача виявлення і виділення стегосообщенія є класичною для теорії зв'язку. Однак такий підхід не враховує двох факторів: невипадкового характеру сигналу контейнера і вимог по збереженню його якості.

Ці моменти не зустрічаються в відомої теорії виявлення і виділення сигналів на фоні адитивного шуму. Їх облік дозволить побудувати більш ефективні стегосистеми.

Розрізняють стегодетектори, призначені для виявлення факту наявності цифрового водяного знаку, і пристрої, призначені для виділення цього цифрового водяного знаку (стегодекодери). У першому випадку можливі детектори з жорсткими (так / ні) або м'якими рішеннями. Для винесення рішення про наявність / відсутність цифрового водяного знаку зручно використовувати такі заходи, як відстань по Хеммінга, або взаємну кореляцію між наявними сигналом і оригіналом. Якщо у нас немає вихідного сигналу, то в справу вступають більш тонкі статистичні методи, засновані на побудові моделей досліджуваного класу сигналів. Залежно від того, яка інформація потрібна детектору для виявлення цифрового водяного знаку, стегосистеми з цифровим водяним знаком діляться на три класи: відкриті, напівзакриті і закриті системи.

Розглянемо докладніше поняття контейнера. До стегокодера - це порожній контейнер, після нього - заповнений контейнер, або Стего. Стего повинен бути візуально не відрізняється від порожнього контейнера. Розрізняють два основних типи контейнерів: потоковий і фіксований.

Потоковий контейнер являє собою безперервно наступну послідовність біт. Повідомлення вкладається в нього в реальному масштабі часу, так що в кодере невідомо заздалегідь, чи вистачить розмірів контейнера для передачі всього повідомлення. В один контейнер великого розміру може бути вбудовано і кілька повідомлень. Інтервали між вбудованими битами визначаються генератором псевдослучайной послідовності з рівномірним розподілом інтервалів між відліками. Основна складність полягає в здійсненні синхронізації, визначенні початку і кінця послідовності.

У фіксованого контейнера розміри і характеристики заздалегідь відомі. Це дозволяє здійснювати вкладення даних оптимальним в деякому сенсі чином. Контейнер може бути обраним, випадковим або нав'язаним. Обраний контейнер залежить від вбудованого повідомлення, а в граничному випадку є його

функцією. Цей тип контейнера більше характерний для стеганографії. Нав'язаний контейнер може з'явитися в сценарії, коли особа, яка надає контейнер, підозрює про можливу прихованої листуванні і бажає запобігти їй. На практиці ж найчастіше стикаються з випадковим контейнером.

Вбудовування повідомлення в контейнер може проводитися за допомогою ключа, одного або декількох. Ключ - псевдослучайная послідовність біт, породжувана генератором, що задовольняє певним вимогам (безпечний криптографічний генератор). Числа, що породжуються генератором псевдослучайної послідовності, можуть визначати позиції модифікуються відліків в разі фіксованого контейнера або інтервали між ними в разі потокового контейнера. Прихована інформація впроваджується відповідно до ключа в ті відліки, спотворення яких не призводить до суттєвих перекручень контейнера. Ці біти утворюють стегопуть. В залежності від програми, під істотним спотворенням можна розуміти спотворення, що приводить як до неприйнятність для людини-адресата заповненого контейнера, так і до можливості виявлення факту наявності прихованого повідомлення після стегоаналіза[2].

Цифрові водяні знаки можуть бути трьох типів: робасні, тендітні і полухрупкіе. Під робастний розуміється стійкість цифрового водяного до різного роду впливів на стегосистем. Робастних цифрових водяних знаків присвячено більшість досліджень. Тендітні цифрові знаки руйнуються при незначній модифікації заповненого контейнера. Вони застосовуються для аутентифікації сигналів. Відмінність від засобів електронного цифрового підпису полягає в тому, що тендітні цифрові водяні знаки все ж допускають деяку модифікацію контенту. Це важливо для захисту мультимедійної інформації, так як законний користувач може, наприклад, побажати стиснути зображення. Інша відмінність полягає в тому, що тендітні цифрові водяні знаки повинні не тільки відобразити факт модифікації контейнера, але також вид і місце розташування цієї зміни. Полухрупкі цифрові водяні знаки стійкі по відношенню до одних впливів і нестійкі по відношенню до інших. Взагалі кажучи, всі можуть бути віднесені до цього типу. Однак полухрупкіе цифрові водяні знаки спеціально проектуються

так, щоб бути нестійкими по відношенню до певного роду операцій. Наприклад, вони можуть дозволяти виконувати стиснення зображення, але забороняти вирізку з нього або вставку в нього фрагмента.

Для того щоб стегосистеми була надійною, при її проектуванні необхідно виконання ряду вимог:

- безпека системи повинна повністю визначатися секретністю ключа. Це означає, що порушник може повністю знати всі алгоритми роботи стегосистеми і статистичні характеристики множин повідомлень і контейнерів, і це не дасть йому ніякої додаткової інформації про наявність чи відсутність повідомлення в даному контейнері;
- знання порушником факту наявності повідомлення в будь-якому контейнері не повинно допомогти йому при виявленні повідомлень в інших контейнерах;
- заповнений контейнер повинен бути візуально не відрізняється від незаповненого. Для задоволення цієї вимоги треба, здавалося б, впроваджувати приховане повідомлення в візуально не значущі області сигналу. Однак, ці ж області використовують і алгоритми стиснення. Тому, якщо зображення буде надалі піддаватися стиску, то приховане повідомлення може зруйнуватися. Отже, біти повинні вбудовуватися в візуально значущі області, а відносна непомітність може бути досягнута за рахунок використання спеціальних методів, наприклад, модуляції з розширенням спектра;
- стегосистеми з цифровим водяним знаком повинна мати низьку ймовірність помилкового виявлення прихованого повідомлення в сигналі, його що не містить. У деяких додатках таке виявлення може призвести до серйозних наслідків. Наприклад, помилкове виявлення цифрового водяного знаку на DVD-диску може викликати відмову від його відтворення плеєром;

- повинна забезпечуватися необхідна пропускна здатність (ця вимога актуальна, в основному, для стегосістем прихованої передачі інформації);
- стегосистеми повинна мати прийнятну обчислювальну складність реалізації. При цьому можлива асиметрична за складністю реалізації система з цифровим водяним знаком, тобто складний стегокодер і простий стегодекодер.

Вимоги до цифрового водяного знаку:

- цифровий водяний знак повинен легко (обчислювальна) вилучатись законним користувачем;
- цифровий водяний знак повинен бути стійким або нестійким до навмисним і випадковим впливам (в залежності про додатки). Якщо цифровий водяний знак використовується для підтвердження автентичності, то неприпустиме зміна контейнера повинно призводити до руйнування цифрового водяного знаку (крихкий цифровий водяний знак). Якщо ж цифровий водяний знак містить ідентифікаційний код, логотип фірми і т.п., то він повинен зберегтися при максимальних викривлення контейнера, звичайно, що не приводять до істотних спотворень вихідного сигналу. Крім того, цифровий водяний знак повинен бути робастних по відношенню до афінних перетворень зображення, тобто його поворотів, масштабування. При цьому треба розрізняти стійкість самого цифрового водяного знаку і здатність декодера вірно його виявити. Наприклад, при повороті зображення цифрового водяного знаку не зруйнується, а декодер може виявитися нездатним виділити його. Існують додатки, коли цифровий водяний знак повинен бути стійким по відношенню до одних перетворень і нестійким по відношенню до інших. Наприклад, може бути дозволено копіювання зображення (ксерокс, сканер), але накладена заборона на внесення в нього будь-яких змін;
- повинна бути можливість додавання до стего-додаткових цифрових знаків. Наприклад, на DVD-диску є мітка про допустимість одноразового

копіювання. Після здійснення такого копіювання необхідно додати мітку про заборону подальшого копіювання. Можна було б, звичайно, видалити перший цифровий водяний знак і записати на його місце другий. Однак, це суперечить припущенню про важко видаляємий цифровий водяний знак. Кращим виходом є додавання ще одного цифрового водяного знаку, після якого перший не братиметься до уваги. Однак, наявність декількох цифрового водяного знаку на одному повідомленні може полегшити атаку з боку порушника.

В даний час можна виділити три тісно пов'язаних між собою і мають одне коріння та напрями докладання стеганографії: приховування даних та повідомлень, цифрові водяні знаки і заголовки[5].

Цифрові водяні знаки можуть застосовуватися, в основному, для захисту від копіювання та несанкціонованого використання. У зв'язку з бурхливим розвитком технологій мультимедіа гостро постало питання захисту авторських прав та інтелектуальної власності, представленої в цифровому вигляді. Прикладами можуть бути фотографії, аудіо і відеозаписи і т.д. Переваги, які дають уявлення і передача повідомлень в цифровому вигляді, можуть виявитися покарбованими легкістю, з якою можливо їх злодійство або модифікація. Тому розробляються різні заходи захисту інформації, організаційного і технічного характеру. Один з найбільш ефективних технічних засобів захисту мультимедійної інформації і полягає у встановленні в об'єкт, що захищається невидимих міток - водяних знаків. Розробки в цій галузі ведуть найбільші фірми в усьому світі. Так як методи цифрових водяних знаків почали розроблятися зовсім недавно, то тут є багато незрозумілих проблем, які вимагають свого вирішення[3].

Переваги, які дають уявлення і передача повідомлень в цифровому вигляді, можуть виявитися покарбованими легкістю, з якою можливо їх злодійство або модифікація. На відміну від звичайних водяних знаків цифрові знаки можуть бути не тільки видимими, але і (як правило) невидимими. Невидимі аналізуються спеціальним декодером, який виносить рішення про їх коректності. Цифрові водяні знаки можуть містити деякий автентичний код, інформацію про власника,

або яку-небудь інформацію, що управляє. Найбільш придатними об'єктами захисту за допомогою цифрових водяних знаків є нерухомі зображення, файли аудіо і відео даних[5].

Технологія вбудовування ідентифікаційних номерів виробників має багато спільного з технологією водяних знаків. Відмінність полягає в тому, що в першому випадку кожна захищена копія має свій унікальний вбудований номер (звідси і назва - дослівно «відбитки пальців»). Цей ідентифікаційний номер дозволяє виробникові відстежувати подальшу долю свого дітища: чи не зайнявся хтось із покупців незаконним тиражуванням. Якщо так, то «відбитки пальців» швидко вкажуть на винного. Вбудовування заголовків (невидиме) може застосовуватися, наприклад, для підпису медичних знімків, нанесення легенди на карту і т.д. Метою є зберігання різнорідно представленої інформації в єдине ціле. Це, мабуть, єдине додаток стеганографії, де в явному вигляді відсутня потенційний порушник[3].

Найбільш істотна відмінність постановки завдання прихованої передачі даних від постановки завдання вбудовування ЦВЗ полягає в тому, що в першому випадку порушник повинен виявити приховане повідомлення, тоді як у другому випадку про його існування всі знають. Більш того, у порушника на законних підставах може бути пристрій виявлення ЦВЗ (наприклад, у складі DVD-програвача)[3].

Основними вимогами, які пред'являються до водяних знаків, є надійність і стійкість до спотворень, вони повинні задовольняти суперечливим вимогам візуальної (аудіо) непомітності і робастности до основних операцій обробки сигналів[3].

Цифрові водяні знаки мають невеликий обсяг, проте, з урахуванням зазначених вище вимог, для їх вбудовування використовуються більш складні методи, ніж для вбудовування просто повідомлень або заголовків. Завдання вбудовування та виділення цифрових водяних знаків з іншої інформації виконує спеціальна стегосистеми[2].

Перш, ніж здійснити вкладення цифрового водяного знаку в контейнер, водяний знак повинен бути перетворений до деякого невластивому увазі. Наприклад, якщо в якості контейнера виступає зображення, то і послідовність ЦВЗ часто представляється як двовимірний масив біт. Для того, щоб підвищити стійкість до спотворень нерідко виконують його завадостійке кодування, або застосовують широкосмугові сигнали. Первісну обробку прихованого повідомлення виконує показаний на рис. 2 прекодер. В якості найважливішої попередньої обробки цифрового водяного знаку (а також і контейнера) назвемо обчислення його узагальненого перетворення Фур'є. Це дозволяє здійснити вбудовування ЦВЗ в спектральної області, що значно підвищує його стійкість до спотворень. Попередня обробка часто виконується з використанням ключа для підвищення секретності вбудовування. Далі водяний знак «вкладається» в контейнер, наприклад, шляхом модифікації молодших значущих біт коефіцієнтів. Цей процес можливий завдяки особливостям системи сприйняття людини. Добре відомо, що зображення мають велику псіховізуальной надмірністю. Око людини подібний до низькочастотному фільтр, що пропускає дрібні деталі. Особливо непомітні спотворення в високочастотної області зображень. Ці особливості людського зору використовуються, наприклад, при розробці алгоритмів стиснення зображень і відео[3].

Процес впровадження цифрових водяних знаків також повинен враховувати властивості системи сприйняття людини. Стеганографія використовує наявну в сигналах псіховізуальную надмірність, але іншим, ніж при стисканні даних чином. Наведемо простий приклад. Розглянемо півтонування з 256 градаціями сірого, тобто з питомою швидкістю кодування 8 біт / піксель. Добре відомо, що око людини не здатний помітити зміну молодшого значущого біта. Ще в 1989 році було отримано патент на спосіб прихованого вкладення інформації в зображення шляхом модифікації молодшого значущого біта. В даному випадку детектор Стего аналізує тільки значення цього біта для кожного пікселя, а очей людини, навпаки, сприймає тільки старші 7 біт. Даний метод простий у реалізації і ефективний, але не задовольняє деяким важливим вимогам до ЦВЗ[5].

У більшості стегосістем для впровадження і виділення цифрових водяних знаків використовується ключ. Ключ може бути призначений для вузького кола осіб або ж бути загальнодоступним. Наприклад, ключ повинен міститися у всіх DVD-плеєрах, щоб вони могли прочитати містяться на дисках ЦВЗ. Не існує, наскільки відомо, стегосистеми, в якій би при виділенні водяного знака була потрібна інша інформація, ніж при його вкладенні[5].

У стегодетекторе відбувається виявлення цифрового водяного знаку в (можливо зміненому) захищеному ЦВЗ зображенні. Ця зміна може бути обумовлено впливом помилок в каналі зв'язку, операцій обробки сигналу, навмисних атак порушників. У багатьох моделях стегосістем сигнал-контейнер розглядається як адитивний шум. Тоді задача виявлення і виділення стегосообщенія є класичною для теорії зв'язку. Однак такий підхід не враховує двох факторів: невипадкового характеру сигналу контейнера і вимог по збереженню його якості. Ці моменти не зустрічаються в відомої теорії виявлення і виділення сигналів на фоні адитивного шуму. Їх облік дозволить побудувати більш ефективні стегосистеми[5].

Розрізняють стегодетектори, призначені для виявлення факту наявності водяного знака і пристрої, призначені для його виділення (стегодекодери). У першому випадку можливі детектори з жорсткими (так / ні) або м'якими рішеннями. Для винесення рішення про наявність / відсутність цифрового водяного знаку зручно використовувати такі заходи, як відстань по Хеммінга, або взаємну кореляцію між наявними сигналом і оригіналом (при наявності останнього, зрозуміло). А що робити, якщо у нас немає вихідного сигналу? Тоді в справу вступають більш тонкі статистичні методи, засновані на побудові моделей досліджуваного класу сигналів[5].

Найбільш затребуваний для захисту авторських прав на фотозображення метод стеганографії - це вбудовування так званих цифрових водяних знаків (ЦВЗ), налаштований на впровадження в мультимедійний файл прихованих маркерів, стійких до різних атак. Вбудовані ЦВЗ аналізуються спеціальним декодером,

виносять рішення про їх коректності. Як ЦВЗ можуть використовуватися дані автора або автентичний код, а також якась інформація, що управляє[5].

До вбудовування інформації в зображення в якості ЦВЗ необхідно перетворити її до потрібного виду, тобто в двовимірний масив біт. Також зазвичай застосовують завадостійке кодування або широкосмугові сигнали для підвищення стійкості цифрового водяного знаку до спотворення. Найважливіша попередня обробка прихованого повідомлення і контейнера - це обчислення їх узагальненого перетворення Фур'є, що дозволяє реалізувати вбудовування ЦВЗ в спектральної області для підвищення його стійкості до спотворень. При попередній обробці повідомлень часто застосовують ключі з метою підвищення секретності вбудовування інформації. Наступний етап - це вже безпосередньо впровадження знаку в зображення, часто з використанням так званого стегоключа (псевдослучайной послідовності біт, отриманої від певного генератора). Непомітність цифрового водяного знаку можлива завдяки великій псіховізуальной надмірності зображень для системи сприйняття людини[5].

Найбільш поширений метод вбудовування починається з вибору стегопутей, тобто біт контейнера, які можна модифікувати без помітних спотворень, після чого за допомогою ключа з них вибираються біти, замінні битами повідомлення. Зазвичай застосовують такі методи, як інверсія, вставка або видалення біта, а також використання порогових біт або різних таблиць значень.

Виділення прихованого в захищуваному зображенні ЦВЗ відбувається в стегодетекторе за допомогою стегоключа, використаного при впровадженні інформації. Цей ключ може бути як загальнодоступним, так і призначеним для вузького кола осіб. Якщо він є загальнодоступним, то такий стегосистеми складно протистояти можливим атакам з боку зловмисників. Крім того, саме захищається зображення може бути зміненим, наприклад, через атаки на нього або операцій обробки сигналу. Результатом таких модифікацій і атак стає неможливість виявлення цифровий водяний знак в стегодетекторе, тому облік цих факторів дозволяє будувати більш ефективні захисні стегосистеми[5].

Стегодетектори діляться на дві категорії. Одні з них призначені для виявлення наявності ЦВЗ, інші - для його виділення (такі детектори називаються стегодекодерамі). У свою чергу, цифрові водяні знаки у стегосистеми діляться на три класи: відкриті, напівзакриті і закриті (мають найбільшу стійкість до зовнішніх впливів). Відповідно, для захисту зображень від незаконного використання найкраще використовувати закриті стегосистеми[5].

Цифрові водяні знаки бувають трьох типів: тендітні, полухрупкіе і робастні. Тендітні ЦВЗ зазвичай руйнуються при зміні стегоконтейнер і застосовуються для аутентифікації сигналів. Робастні ЦВЗ стійкі до різних видів впливу на контейнер, тому саме цього типу міток присвячено безліч розробок. Полухрупкіе ЦВЗ зазвичай бувають стійкими до одного виду впливів, але нестійкими по відношенню до інших. Строго кажучи, робастні ЦВЗ теж можна віднести до полухрупкім, оскільки абсолютної стійкості міток домогтися дуже складно. Для захисту цифрових фотографій краще використовувати робастні ЦВЗ, оскільки у зображень можуть бути відредаговані такі параметри, як колірна гамма, яскравість, воно може бути отмасштабовані або повернуто. Крім того, такі цифрові водяні знаки повинні виявлятися тільки однією стороною, щоб зловмисник не міг їх виділити і знищити[5].

Цілком логічно, що захищена ЦВЗ фотографія повинна бути візуально не відрізняється від вихідної. Найпростіше вбудовувати інформацію в незначні біти зображення (LSB-метод), що внесення змін до молодших бітів в більшості випадків не викликає значної трансформації зображення і не виявляється - 55 - візуально. Але оскільки ці ж області використовуються алгоритмами стиснення, то при такого роду деформації ЦВЗ буде руйнуватися[5].

Існують стегосистеми, в яких розділені етапи виявлення і аутентифікації, в результаті чого цифровий водяний знак легко виявляється, але видалити його непросто. Такі системи будуються на основі цифрового водяного знаку з нульовим знанням і цілком можуть бути використані для захисту цифрових фотографій[5].

Сучасний підхід до захисту цифрових фотографій з використанням ЦВЗ полягає у вбудовуванні інформації в ті області зображення, руйнування яких призведе до його необоротної псування. При цьому стегоалгоритму враховують як стиснення зображень, так і властивості людського зору[5].

1.4. Переваги та недоліки методів контролю цілісності, які ґрунтуються на використанні цифрових водяних знаків

Істотним недоліком використання контролю цілісності з використання цифрового водяного знаку є те, що водяні знаки можуть бути знайдені за допомогою статичного аналізу коду або спотворені рівномірними обфускація, що вони бережуть семантику програми.

Так само перевагою є те що не можливо на перший погляд визначити що зображення схильне контролю цілісності. Цифровий водяний знак непомітний на зображенні, а змінюючи молодший біт кольору в пікселі неможливо визначити людським оком[3].

1.5. Висновок

В даному розділі були розглянуті методи контролю цілісності зображень. Більш детально розглянемо методи контролю цілісності, які ґрунтуються на використанні цифрового водяного знаку. Більш докладно розглянемо цифрові водяні знаки. Так само озвучені недоліки і гідності методу контролю цілісності на основі використання цифрового водяного знаку.

2. РОЗРОБКА МЕТОДУ КОНТРОЛЮ ЦІЛІСНОСТІ ЗОБРАЖЕННЯ НА ОСНОВІ ЦИФРОВОГО ВОДЯНОГО ЗНАКА

Як було зазначено в попередньому розділі, для кращого контролю цілісності зображень краще використовувати метод, який заснований на впровадженні хеш-суми як цифровий водяний знак, так як він не тільки вирішує проблему контролю цілісності зображень, а також приховує той факт, що зображення схильне контролю цілісності.

У цьому розділі буде розглянуто метод контролю цілісності зображень з впровадженням цифрового водяного знаку на основі алгоритму стиснення зображення формату «JPEG» та хеш-функції. Спочатку необхідно провести експеримент, на основі якого можна буде зробити висновки про проблему методу на основі стиснення даних. Він полягає у розробці методу вбудови хеш-суми у квантовані компоненти зображення на етапах стиснення. Виконується аналіз та виявлення найбільш сприятливих компонент дискретного косинусного перетворення на основі показників стиснення компоненту та його власного значення.

2.1 Хеш-функція SHA1

Secure Hash Algorithm 1 - алгоритм криптографічного хешірованія. Для вхідного повідомлення довільної довжини алгоритм генерує 160-бітове (20 байт) хеш-значення, зване також дайджестом повідомлення, яке зазвичай відображається як шестнадцатиричное число, довжиною в 40 цифр[6].

SHA-1 реалізує хеш-функцію, як функцію стиснення. Входами функції стиснення є блок повідомлення довжиною 512 біт (8x8 матриця - M), і вихід попереднього блоку. Вихід є значення всіх хеш-блоків до цього моменту. Іншими словами хеш-блок $h_i = f(M_i, h_{i-1})$. Хеш-значення всього повідомлення вихід останнього блоку (h_i)[6].

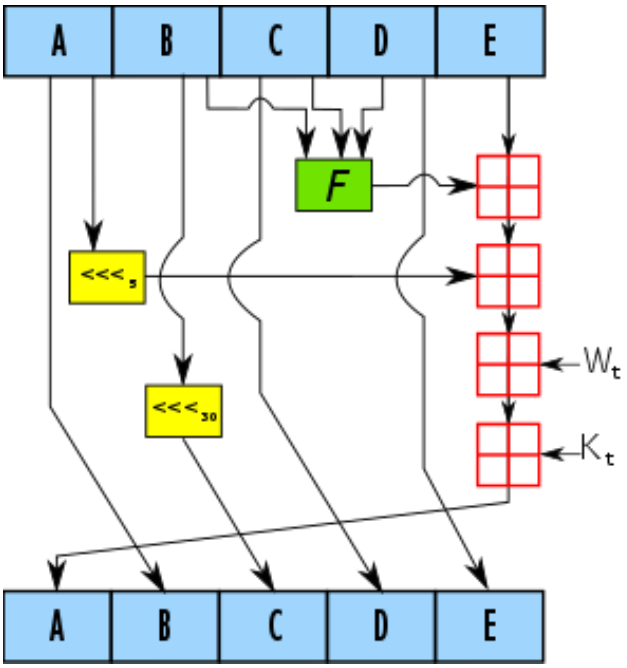

Рисунок 2.5 Одна ітерація алгоритму SHA1

Визначаються чотири нелінійні операції і чотири константи (рис. 2.6):

$F_t(m,l,k) = (m \wedge l) \vee (\neg m \wedge k)$	K_t = 0x5A827999	0≤t≤19
$F_t(m,l,k) = m \oplus l \oplus k$	K_t = 0x6ED9EBA1	20≤t≤39
$F_t(m,l,k) = (m \wedge l) \vee (m \wedge k) \vee (l \wedge k)$	K_t = 0x8F1BBCDC	40≤t≤59
$F_t(m,l,k) = m \oplus l \oplus k$	K_t = 0xCA62C1D6	60≤t≤79

Рисунок 2.6 Операції та константи SHA1

В методі будемо використовувати 160 бітів, так як ця кількість є найбільш оптимальним:

1) Складність генерування хеш-суми буде ще на кілька років неприступною для підбору хеш-суми, так як при розв'язанні методом «грубої сили»: завдання знаходження колізій - ситуація, коли двом різним вихідним повідомленнями відповідає одне і те ж хеш-значення вимагає $2^{160/2} = 2^{80}$ операцій, а завдання знаходження прообразу - початкового повідомлення - по його хешу, вимагає 2^{160} операцій (рис. 2.7)[6].

2) Впроваджувальна кількість бітів є оптимальною, тому що лише 160 бітів будуть змінені і це буде збільшувати помилку другого роду для статичних аналізаторів, так як вони залежать від кількості змінених бітів у зображенні[6].

Function	Collision type	GPU	Time	Complexity	Cost
SHA-1	collision	GTX 970	22 years	$2^{61.2}$	
		GTX 1060	27 years	$2^{61.6}$	11k US$
		GTX 1080 Ti	8 years	$2^{61.6}$	
	chosen-prefix	GTX 970	99 years	$2^{63.4}$	
		GTX 1060	107 years	$2^{63.5}$	45k US$
		GTX 1080 Ti	34 years	$2^{63.6}$	
MD5 ‖ SHA-1	both (plain or CP)	GTX 970	1400 years	$2^{67.2}$	
		GTX 1060	1700 years	$2^{67.6}$	720k US$
		GTX 1080 Ti	540 years	$2^{67.6}$	

Рисунок 2.7 Складність атак на SHA-1

2.2 Розробка математичного забезпечення стеганографічного методу

JPEG – один з нових і досить потужних алгоритмів. Оперує алгоритм областями 8x8, на яких яскравість міняється порівняно плавно. Внаслідок цього при розкладанні матриці такий, області в подвійний ряд по косинусах значимими виявляються тільки перші коефіцієнти. Алгоритм розроблений групою експертів в області фотографії. JPEG – Joint Photographic Expert Group – підрозділ у рамках ISO – Міжнародній організації по стандартизації[7].

Основними етапами стиснення зображення вибраного формату є:

1. Переклад простір RGB в колірний простір YCbCr.
2. Субдискретизація компонент кольоровості.
3. Застосування дискретного косинусного перетворення.
4. Квантування.
5. "Зигзаг"-сканування.
6. Алгоритм групового кодування (RLE).

7. Кодування за Хаффманом.

Створений метод використовує операції перекладу зображення з простору RGB у простір YCbCr, дискретне косинусне перетворення та квантування.

Переклад простір RGB в колірний простір YCbCr.

В цьому просторі Y – компонента яскравості, Cb і Cr – компоненти кольоровості. Людське око більше чутливе до яскравості, ніж до кольору (рисунок 2.8). Тому важливіше зберегти більшу точність при передачі Y, ніж при передачі Cb і Cr.

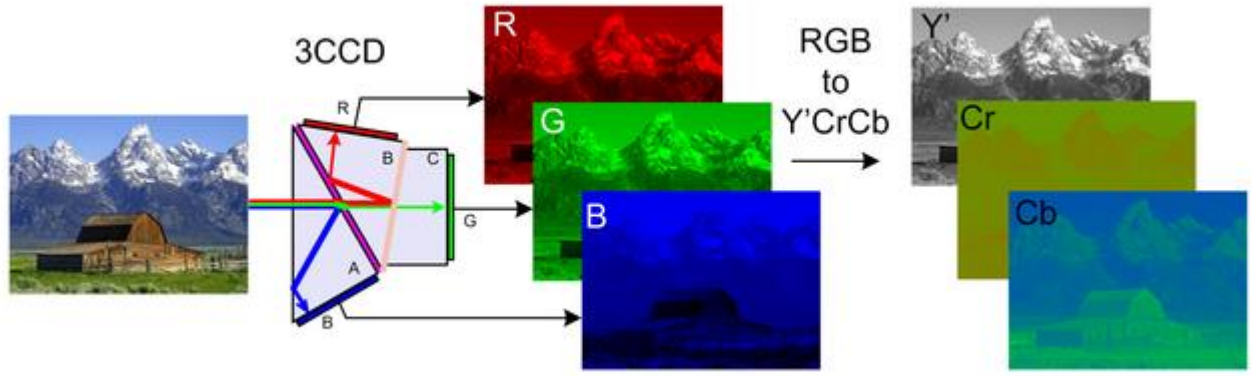

Рисунок 2.8 – Переклад простір RGB в колірний простір YCbCr.

Переклад здійснюється за такою формулою:

$$Y' = K_R \cdot R' + K_G \cdot G' + K_B \cdot B'$$
$$P_B = \frac{1}{2} \cdot \frac{B' - Y'}{1 - K_B}$$
$$P_R = \frac{1}{2} \cdot \frac{R' - Y'}{1 - K_R}$$

Де K_R, K_G та K_B зазвичай походять із визначення відповідного простору RGB і вимагають задоволення $K_R + K_G + K_B = 1$. Еквівалентна маніпуляція матрицею часто називають "кольоровою матрицею"[8]:

$$\begin{bmatrix} Y' \\ P_B \\ P_R \end{bmatrix} = \begin{bmatrix} K_R & K_G & K_B \\ -\frac{1}{2} \cdot \frac{K_R}{1-K_B} & -\frac{1}{2} \cdot \frac{K_G}{1-K_B} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \cdot \frac{K_G}{1-K_R} & -\frac{1}{2} \cdot \frac{K_B}{1-K_R} \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

Застосування дискретного косинусного перетворення.

В алгоритмі JPEG використовується дискретне косинусне перетворення, яке розкладає зображення по амплітудам деяких частот. Таким чином, при перетворенні виходить матриця, в якій більшість коефіцієнтів або близькі, або дорівнюють нулю. Крім того, завдяки недосконалості людського зору можна апроксимувати коефіцієнти більш грубо без помітної втрати якості зображення. Для цього використовується квантування коефіцієнтів. У найпростішому випадку – це арифметичне побітове зрушення вправо[9]. При цьому перетворенні втрачається частина інформації, але може досягатися великий ступінь стиснення. Порядок роботи прямого і зворотного алгоритмів наведені на рисунку 2.9.

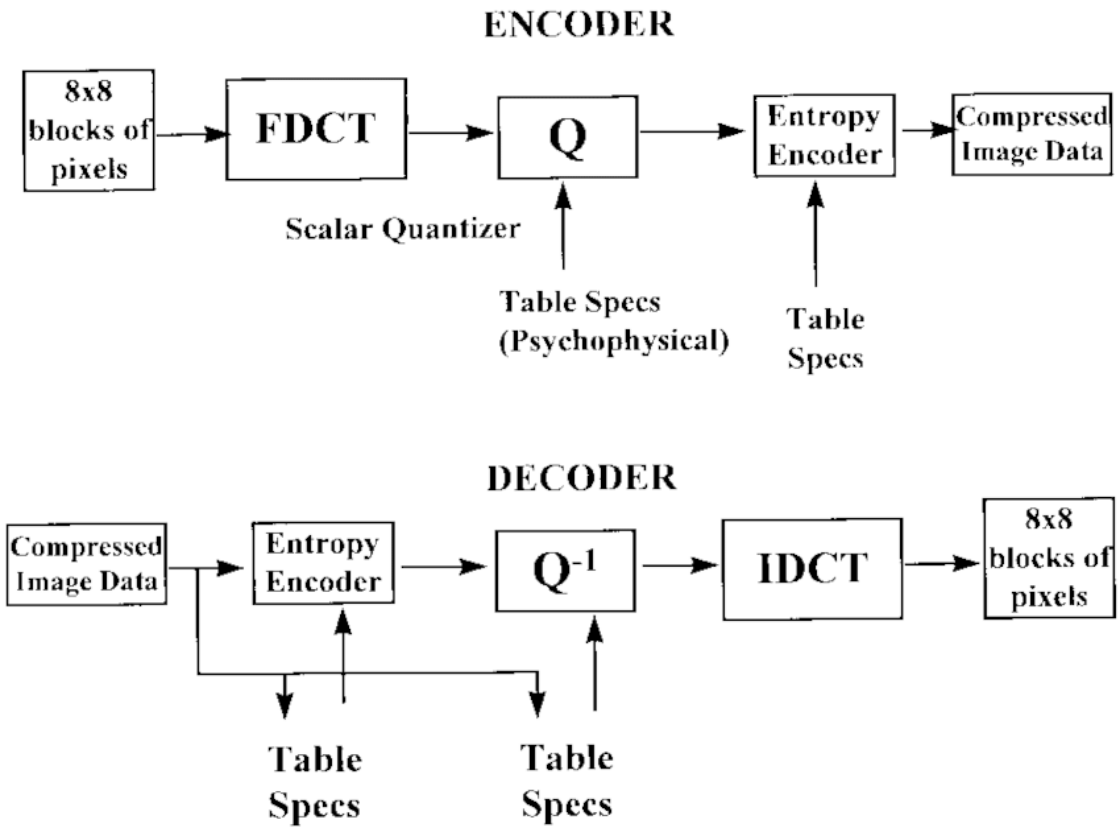

Рисунок 2.9 – Схема стиснення та відновлення зображень по JPEG

Для перетворення використовуються наступні формули:

Формула прямого дискретного косинусного перетворення:

$$ДКП(\text{i,j}) = \frac{1}{\sqrt{2N}} \times C(i) \times C(j) \times \sum_{x=0}^{\text{N-1}} \sum_{y=0}^{\text{N-1}} \text{pixel(x,y)} \times \cos\left[\frac{(2\text{x}+1)\text{dx}}{2N}\right] \times$$

$$\times \cos\left[\frac{(2\text{y}+1)\text{dx}}{2N}\right]$$

$$C(x)\begin{cases}\frac{1}{\sqrt{2}}, x = 0\\ 1, \text{x}>0\end{cases}$$

Формула зворотного дискретного косинусного перетворення:

$$ДКП(\text{i,j}) = \frac{1}{\sqrt{2N}} \times\times \sum_{x=0}^{\text{N-1}} \sum_{y=0}^{\text{N-1}} C(i) \times C(j)_{ДКП}(\text{i,j}) \times \cos\left[\frac{(2\text{x}+1)\text{ix}}{2N}\right] \times$$

$$\times \cos\left[\frac{(2\text{y}+1)\text{jx}}{2N}\right]$$

$$C(x)\begin{cases}\frac{1}{\sqrt{2}}, x = 0\\ 1, \text{x}>0\end{cases}$$

У отриманій матриці коефіцієнтів, низькочастотні компоненти розташовані ближче до лівого верхнього кута, а високочастотні – справа і внизу. Це важливо тому, що більшість графічних образів на екрані комп'ютера складається з низькочастотної інформації. Високочастотні компоненти не так важливі для передачі зображення[10].

Для оптимізації виконання дискретного косинусного перетворення, зображення розбивається на блоки розміром 8x8 точок.

Значно більш ефективний варіант обчислення коефіцієнтів дискретного косинусного перетворення реалізований через перемноження матриць.

При такому підході формула дискретного косинусного перетворення може бути записана в наступному вигляді:

Пряме:

$$\text{ДКП} = \text{С} \times (\text{Точки} \times \text{С}_\text{Т}) \qquad (2.1)$$

Зворотне:

$$\text{Точки} = \text{С}_\text{Т} \times (\text{ДКП} \times \text{С}) \qquad (2.2)$$

де:

- ДКП – дискретне косинусне перетворення;
- С – матриця косинусного перетворення розміром 8x8, елементи якої визначаються за формулою:

$$C(i,j) = \begin{cases} {}^{1}/_{\sqrt{8}}, \text{при } i = 0 \\ \left(\sqrt{{}^{2}/_{8}}\right) \times \cos((2j+1) \times i \times \pi), \text{при } i > 0 \end{cases}$$

- Точки – матриця розміром 8x8, що складається з пікселів зображення;
- $\text{С}_\text{Т}$ – транспонована матриця.

Потім розглядається матриця С1 і за допомогою тригонометричних перетворень, процедур розкладання матриць на множники приводиться до простішого вигляду[11].

Дискретне косинусне перетворення являє собою перетворення інформації без втрат і не здійснює ніякого стиснення. Дискретне косинусне перетворення готує інформацію для етапу стиснення з втратами або округлення.

Квантування.

Стандарт JPEG реалізує цю процедуру через матрицю округлення. Для кожного елемента матриці дискретного косинусного перетворення існує відповідний елемент матриці округлення. Результуюча матриця виходить розподілом кожного елемента матриці дискретного косинусного перетворення на відповідний елемент матриці округлення і наступним округленням результату до найближчого цілого числа. Як правило, значення елементів матриці округлення ростуть у напрямку зліва направо і зверху вниз.

Вибір матриці округлення. На цьому кроці здійснюється управління ступенем стиснення і відбуваються найбільші втрати. Зрозуміло, що, задаючи матриці квантування з великими коефіцієнтами, виходить більше нулів і, отже, велика ступінь стиснення. Від вибору матриці округлення залежить баланс між ступенем стиснення зображення і його якістю після відновлення. Стандарт JPEG дозволяє використовувати будь-яку матрицю округлення, однак ISO розробила набір матриць округлення[12].

Один з варіантів отримання матриці округлення: за допомогою дуже простого алгоритму. Для того щоб визначити крок зростання значень в матриці округлення, задається одне значення в діапазоні [1, 25], зване фактором якості QualityFactor.

Фактор якості задає інтервал між сусідніми рівнями матриці округлення, розташованими на її діагоналях. Приклад, отриманої таким чином матриці округлення, представлений на рисунок 2.10.

3	5	7	9	11	13	15	17
5	7	11	13	15	17	19	21
7	11	13	15	17	19	21	23
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

Рисунок 2.10 – Матриця округлення з фактором якості, рівним 2.

Другий з варіантів отримання матриці округлення, це Матриця округлення basic_table зі стандарту JPEG. Саме матриці квантування стандарту IJG використовуються у даній роботі (рисунок 2.11).

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	58	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

яскравість

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

кольоровість

Рисунок 2.11 – Матриці квантування

Операція округлення є єдиною фазою роботи JPEG, де відбувається втрата інформації.

2.2. Розробка методу контролю цілісності зображення на основі цифрового водяного знака.

Щоб зробити факт впровадження хеш-суми найменш помітним з точки зору психовізуальної моделі JPEG, необхідно вибрати для впровадження такі ДКП-компоненти блоків зображення, які будуть найменш помітні для людини.

Для визначення міри непомітності зміни компонента, для кожного компонента розраховується значення ефективності впровадження S.

Для підрахунку значення S використовуються міркування - впровадження найменш помітне в тому частотному компоненті, який найменш помітний для людського зору.

Для того, щоб визначити, які частотні компоненти найменш помітні, використовується психовізуальна модель JPEG, зокрема, матриці квантування IJG. Високі числа в матриці квантування відповідають частотним компонентам, які з точки зору психовізуальної моделі JPEG є найменш цінними для сприйняття, тобто, найменш помітними. Впровадження в такі компоненти приведе до менших психовізуальних спотворень. Згідно переважній більшості матриць квантування JPEG, найменш помітними є високочастотні компоненти ДКП[13].

На базі матриці квантування IJG, створюється матриця S, що зіставляє кожному з 64 спектральних компонентів ДКП індекс помітності цього компонента S.

На основі матриці квантування, ми створюємо матрицю індексів, в якій зіставляємо кожному з компонентів ДКП індекс прихованості цього компонента. Таким чином ми отримаємо вектор з цих індексів і їх розташування, вибираємо 160 елементів з найбыльшими індексами прихованості, щоб записати хеш-суму.

Основні кроки розробленого методу контролю цілісності зображення:

Крок 1. Перехід з простору RGB в простір YCrCb.

Крок 2. Поділ зображення на блоки - 8x8 пікселів.

Крок 3. До кожного блоку застосовується двовимірне дискретне косинусное перетворення. В результаті виходить речовинний спектр.

Крок 4. Сформовані блоки піддаються квантованию. Далі з них формується багатовимірний масив.

Таким чином, беруться всі спектральні компоненти всіх блоків зображення, і для кожного з них знаходиться значення прихованості. Отриманий список сортується і вибираються 160 блоків з найбільшими значення.

Крок 5. Вирізаємо вибрані блоки з зображення, пам'ятаючи їх місце розташування та черговність блоків в зображенні і після цього отримуємо зображення без цих блоків, яке пропускаємо через хеш-функцію для знаходження хеш-суму зображення.

Крок 6. Знаючи черговість блоків записуємо в кожен молодший значущий біт блока значення одного з біта хеш-суми. Таким чином ми впроваджуємо хеш-суму в зображення в найбільш непомітні місця.

Крок 7. Додаємо назад змінені вирізані блоки в зображення.

Основні кроки можна побачити на рис. 2.12:

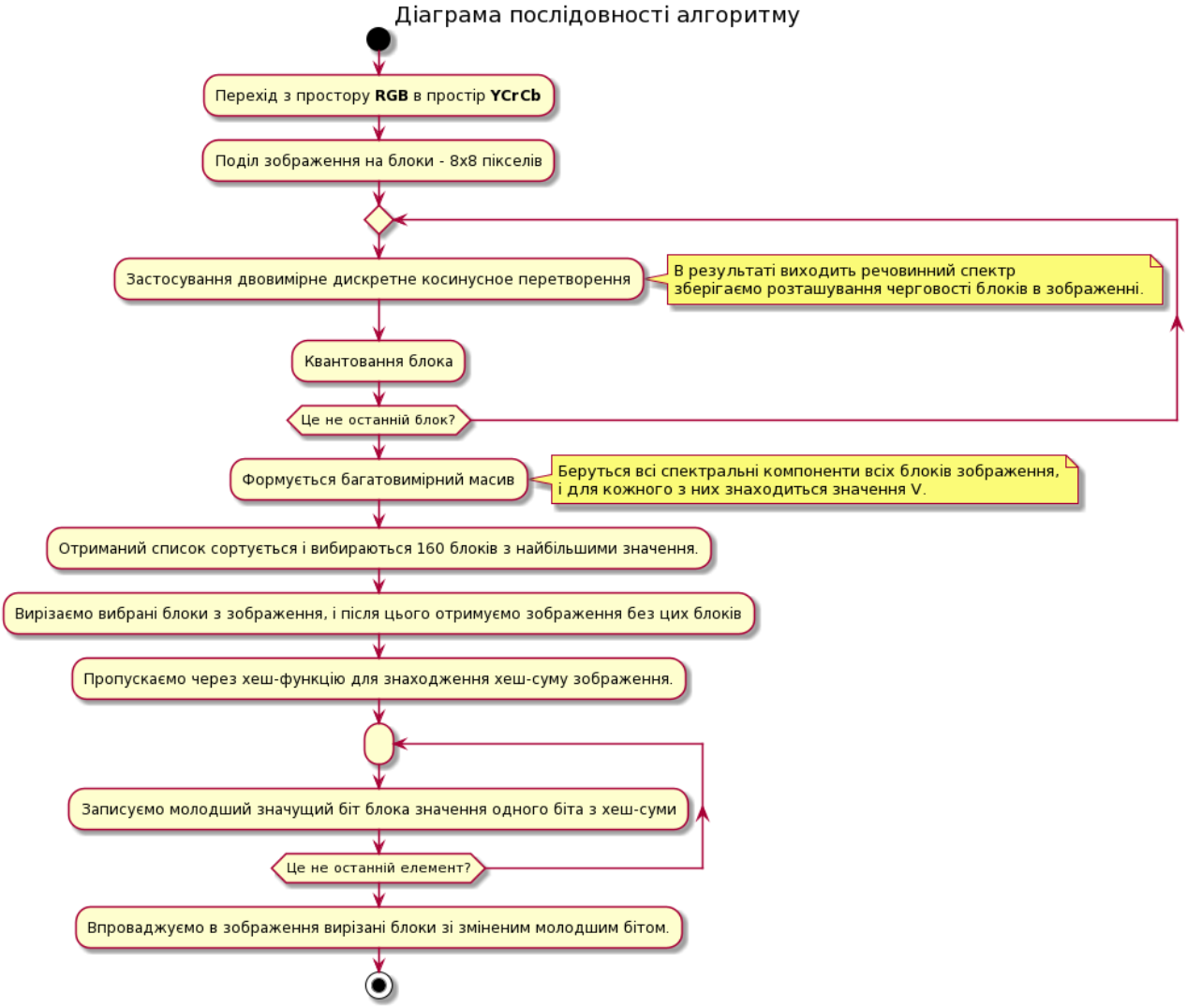

Рис. 2.12 Алгоритм методу контролю цілісності зображення.

2.3. Схема програмного забезпечення

Для кращої продуктивності потрібно поділити програму на блоки, в якому кожен блок буде по етапно, або паралельно в залежності від операцій, які будуть читати, або вирізати дані.

Для реалізації програмного забезпечення буде розглядатися наступні схеми:

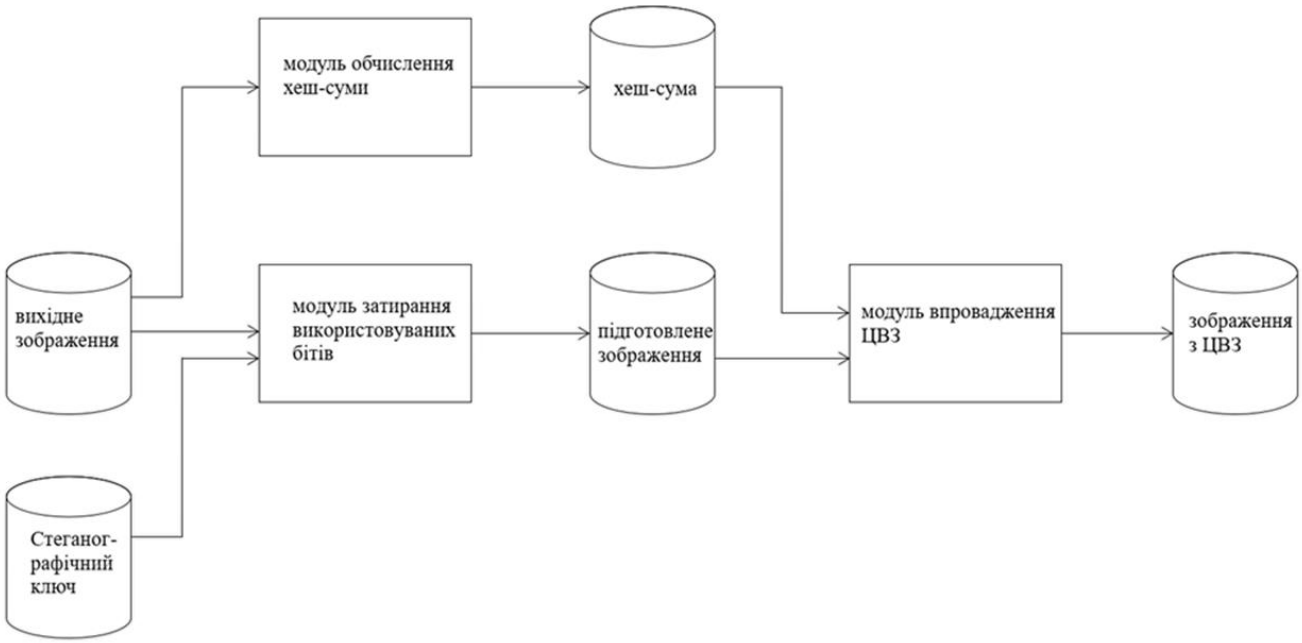

Рисунок 2.13. - Схема потоку даних методу контролю цілісності

На рисунку 2.13. зображена схема потоку даних запропонованого методу контролю цілісності. На даній схемі можна побачити три головних модуля:

- Перший модуль затирання використовуваних бітів, на його вхід подається оригінальне зображення і стенографічний ключ, стенографічний ключ дозволяє визначити стенографічний шлях, який і затирає даний модуль, на виході виходить зображення з уже затертим стенографічним шляхом.
- Другий модуль обчислення хеш-суми, на вхід подається вже змінене зображення для обчислення хеш-суми на виході отримуємо вже готову хеш-суму.
- Третій модуль вбудовування цифрового водяного знаку (ЦВЗ), на вхід подається змінене зображення, хеш-сума і стенографічний ключ, на виході виходить схильне контролю цілісності зображення з цифровим водяним знаком.

Всі ці модулі реалізують метод для того щоб піддати зображення контролю цілісності з подалі перевіркою на зміни.

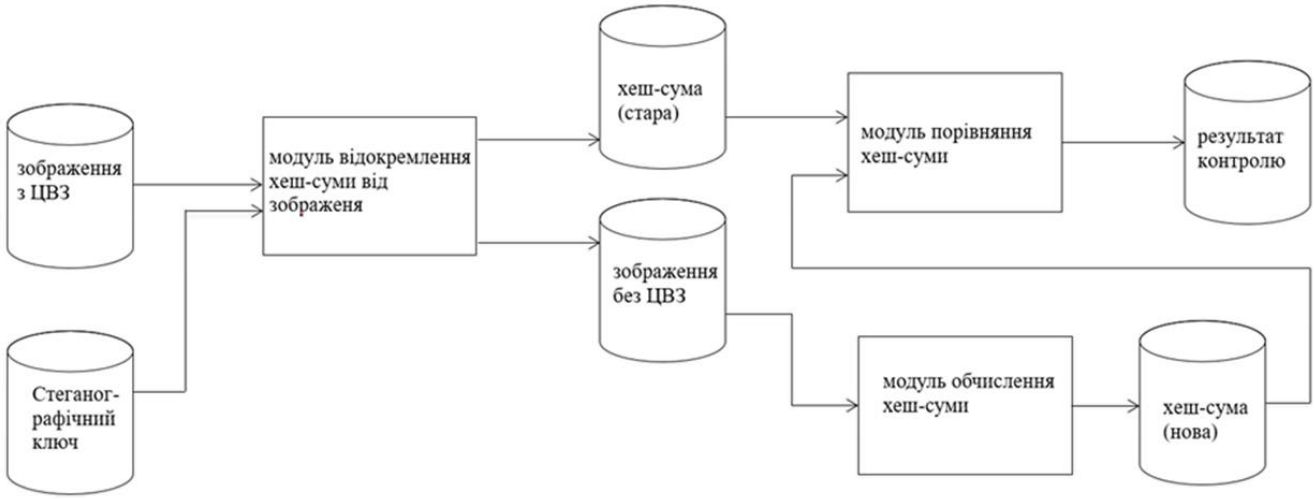

Рисунок 2.14. - Схема потоку даних в зворотному напрямку методу контролю цілісності

На рисунку 2.14. зображена схема потоку даних запропонованого методу контролю цілісності в зворотному напрямку. На даній схемі можна побачити три головних модуля:

- на перший модуль відділення хеш-суми від зображення на вхід подається зображення з цифровим водяним знаком яке попередньо схильне контролю цілісності даним методом, а так само стенографічний ключ для визначення місця в якому записаний цифровий водяний знак на виході це модуль надає хеш-суму яка зберігалася в зображенні у вигляді цифрового водяного знаку, а так само підготовлене зображення з затертими битами де зберігався цифровий водяний знак.
- другий модуль обчислення хеш-суми приймає на вхід оброблене зображення для якого обчислює нову хеш-суму, а на виході видає цю хеш-суму.
- третій модуль порівняння хещ-сум порівнює нову і стару хеш-суму між собою і видає результат порівняння. За умови збігу хеш-сум даний метод видає підтвердження цілісності зображення.

Всі ці модулі реалізують перевірку растрового зображення на зміни навіть якщо були зроблені найменші зміни в зображенні в плоть до одного біта метод це покаже.

2.4. Висновок

В даному розділі було розглянуто хеш-функція, а також було розроблено математичне забезпечення стеганографічного методу. В ході якого були детально розглянуто алгоритм JPEG та всі його його етапи, та був розроблений метод контролю цілісності, після чього також була розроблена схема програмного забезпечення.

3. РЕАЛІЗАЦІЯ МЕТОДУ КОНТРОЛЮ ЦІЛІСНОСТІ ЗОБРАЖЕННЯ НА ОСНОВІ ЦИФРОВОГО ВОДЯНОГО ЗНАКА

3.1 Обґрунтування вибору засобів розробки програмного забезпечення

Для реалізації програмного продукту було обрано інтерпретовану об'єктно-орієнтовану мову програмування високого рівня зі строгою динамічною типізацією – Python.

Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднування наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована[14].

Для розробки та написання програмного коду використовувалися такі бібліотеки як: PyQt5, NumPy та PIL.

PyQT — оболонка на мові програмування Python для бібліотеки Qt. PyQt практично повністю реалізує можливості Qt. А це понад 600 класів, більше 6000 функцій і методів, включаючи:

- існуючий набір віджетів графічного інтерфейсу;
- стилі віджетів;
- доступ до баз даних за допомогою SQL;
- QScintilla, заснований на Scintilla віджет текстового редактора;
- підтримку інтернаціоналізації (i18n);
- парсер XML;
- підтримку SVG;
- інтеграцію з WebKit, движком рендеринга HTML;

- підтримку відтворення відео і аудіо.

Основні модулі програми:

- QtCore – головне не графічні класи: система сигналів і слотів, платформонезавісимость абстракції для Unicode, потоків, що розділяється пам'яті, регулярних виразів і т. д.
- QtGui – компоненти графічного інтерфейсу (елементи управління), засновані на візуальному представленні.
- QtNetwork – класи для мережевого програмування. Наприклад, клієнтів і серверів через UDP і TCP.
- QtOpenGL – класи, що дозволяють використовувати OpenGL і 3D-графіку в додатках PyQt.
- QtScript – класи, що дозволяють використовувати вбудований в Qt інтерпретатор JavaScript для управління додатком.
- QtSql – класи для інтеграції з базами даних за допомогою SQL.
- QtSvg – класи для відображення векторної графіки в форматі SVG.
- QtXml – класи, що реалізують обробку XML.
- uic – реалізація обробки XML-файлів, створених в Qt Designer, для генерації з них Python-коду графічного інтерфейсу[15].
- NumPy — розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами[16].
- Python Imaging Library (PIL) — open-source бібліотека мови Python (версія 2), призначена для роботи з растровою графікою.

Можливості бібліотеки:

- підтримка бінарних, напівтонових, індексованих, повнокольорових і CMYK зображень;
- підтримка форматів BMP, EPS, GIF, JPEG, PDF, PNG, PNM, TIFF і деяких інших у режимі читання та запису;

- підтримка форматів (ICO, MPEG, PCX, PSD, WMF та інших) тільки для читання;
- перетворення зображень з одного формату у інший;
- редагування зображень (використання різноматніх фільтрів, масштабування, малювання, матричні операції і т.п.);
- використання бібліотеки з Tkinter та PyQt.

Python з пакетами NumPy, SciPy і MatPlotLib активно використовується як універсальне середовище для наукових розрахунків в якості заміни поширеним спеціалізованим комерційним пакетам Matlab, IDL і іншим[17].

3.2 Розробка програмного забезпечення

Основні кроки реалізації розробленого методу впровадження повідомлення в контейнер зображення формату JPEG (типи класів системи і різного роду статичні зв'язки, які існують між її складовими представлені у діаграмі класів у додатку Б).

1. Зчитування зображення у RGB просторі у змінну та перехід у простір `YCbCr`.

Для цієї операції потрібно розробити функцію, що буде повертати зображення у вигляді матриці, на вхід буде подаватися місце розташування зображення, а на вихід будемо отримувати 3 матриці розміром таким самим як зображення. В кожній матриці будуть значення одного з кольорових просторів:

```
import numpy as np
from PIL import Image

def jpeg_to_ycbcr(jpeg_path):
     im = Image.open(jpeg_path)
     im = im.convert('YCbCr')
     return np.array(im)
```

2. Поділ зображення на октети – блоки 8x8 (рисунок 3.1).

Для реалізації поділу створюється функція-ітератор, яка буде видавати елементи зображення у вигляді октетів. На вхід буде подаватися зображення у вигляді 3 матриць розміром таким самим як зображення, в кожній матриці будуть значення одного с кольорових просторів:

```
def octet_iterator(ycbcr_image, block=8):
     i = (len(ycbcr_image)//block)
     j = (len(ycbcr_image[0])//block)
     for row in range(i):
          for column in range(j):
          yield [row, column, ycbcr_image[(row*block):((row+1)*block),
     (column*block):((column+1)*block)]]
```

208	213	186	207	210	238	252	227
183	209	194	255	213	225	212	217
197	218	210	255	211	216	203	216
220	223	225	222	215	220	224	210
215	212	225	221	226	228	224	206
216	204	213	243	221	228	214	233
219	200	212	236	219	228	217	241
206	188	224	210	229	229	223	209

Рисунок 3.1 – Приклад октету зображення

3. Застосування дискретного косинусного перетворення для кожного октету (рисунок 3.2).

Далі потрібно написати метод для дискретного косинусного перетворення. Метод буде отримувати на вхід октет, та повертати октет з перетвореними значеннями:

```
def octet_dct(octet):
     octet=octet.astype(np.float64)
     dct_oct = np.zeros(octet.shape)
     def dct2(block):
          return dct(dct(block.T, norm='ortho').T, norm='ortho')

     for i in range(0, 3): dct_mas = dct2(octet[:, :, i])
          for row in range(8):
               for column in range(8):
                    dct_oct[row][column][i] = dct_mas[row, column]
     return dct_oct
```

1744,125	-39,9224	-32,1617	0,567686	9,875	19,54604	-0,88459	-29,5427
-9,86804	-6,31548	2,563221	-1,02218	-11,3894	-1,44311	-30,3003	-27,9221
-10,527	-35,0328	7,697146	13,00727	-0,16332	1,530387	-1,03661	-0,36969
13,36689	-17,8007	20,96233	14,67412	1,27325	1,138417	-1,03829	-0,78484
-0,375	1,124016	17,25791	27,55149	-33,125	0,359806	0,834183	37,19153
12,06274	-0,06872	27,77598	1,707329	1,203018	0,505703	0,437121	0,141695
-0,91626	0,700053	1,213388	0,615176	-0,06765	-0,04681	0,802854	1,136339
-0,02927	0,146834	-0,47253	1,000722	-0,52494	-1,16863	0,1737	0,635659

Рисунок 3.2 – Октети зображення після дискретного косинусного перетворення

4. Квантування октетів на основі стандартних матриць квантування JPEG (рисунок 3.3).

Для даної операції необхідно задати матриці для квантування, для цього створюється метод-агрегатор, який буде повертати матрицю квантування:

```
def quantization_tables(Q):
     jpeg_y = np.array([[16, 11, 10, 16, 24, 40, 51, 61],
                        [12, 12, 14, 19, 26, 58, 60, 55],
                        [14, 13, 16, 24, 40, 57, 69, 56],
                        [14, 17, 22, 29, 51, 87, 80, 62],
                        [18, 22, 37, 56, 68, 109, 103, 77],
                        [24, 35, 55, 64, 81, 104, 113, 92],
                        [49, 64, 78, 87, 103, 121, 120, 101],
                        [72, 92, 95, 98, 112, 100, 103, 99]])
     jpeg_c = np.array([[17, 18, 24, 47, 99, 99, 99, 99],
                        [18, 21, 26, 66, 99, 99, 99, 99],
                        [24, 26, 56, 99, 99, 99, 99, 99],
                        [47, 66, 99, 99, 99, 99, 99, 99],
                        [99, 99, 99, 99, 99, 99, 99, 99],
                        [99, 99, 99, 99, 99, 99, 99, 99],
                        [99, 99, 99, 99, 99, 99, 99, 99],
                        [99, 99, 99, 99, 99, 99, 99, 99]])
     if Q<50:
          S = 5000/Q
     else:
          S = 200 -2*Q
     Y = ((jpeg_y*S +50)/100).round()
     Y = np.maximum(Y,np.ones(np.shape(jpeg_y)))
     C = ((jpeg_c*S +50)/100).round()
     C = np.maximum(C,np.ones(np.shape(jpeg_c)))
     return Y, C
```

Наступною потрібно написати функцію для виконання квантування та зворотного квантування. На вхід методу квантування буде подаватися октет після дискретного косинусного перетворення, та повертатися буде кантований октет:

```
def jpeg_div(dct_oct):
     jpeg_y, jpeg_c = quantization_tables(90)
     div_oct = np.zeros(dct_oct.shape)
     for i in range(1, 3):
          div_oct[:, :, i] = dct_oct[:, :, i] / jpeg_c
     div_oct[:, :, 0] = dct_oct[:, :, 0] / jpeg_y
     return np.round(div_oct)
```

581	-20	-16	0	2	2	0	-2
-5	-3	1	0	-2	0	-3	-3
-4	-12	3	3	0	0	0	0
4	-6	5	2	0	0	0	0
0	0	2	3	-2	0	0	2
2	0	3	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Рисунок 3.3 – Октет зображення після квантування

5. Пошук найкращих значень на основі матриць квантування S (рисунок 3.4).

Створюється метод для підрахунку S значення. На вхід він буде отримувати октет та коефіцієнт значення частотною непомітності компоненту, а повертати – октет значень рейтингу:

```
def get_Shf(matrix, q=1):
     matrix = np.array(matrix)
     jpeg_y, jpeg_c = quantization_tables(90)
     max = np.max(jpeg_y)
     min = np.min(jpeg_y)
     for h in range(len(jpeg_y)):
          for w in range(len(jpeg_y[0])):
               if max-min == 0:
                    matrix[h, w, 0] = 0
               else:
                    matrix[h, w, 0] = (jpeg_y[h, w] - min) * (1.0/(max-
               min))
     max = np.max(jpeg_c)
     min = np.min(jpeg_c)
     for h in range(len(jpeg_c)):
```

```
            for w in range(len(jpeg_c[0])):
                if max-min == 0:
                    matrix[h, w, 1] = 0
                    matrix[h, w, 2] = 0
                else:
                    matrix[h, w, 1] = (jpeg_c[h, w] - min) * (1.0/(max-
                min))
                    matrix[h, w, 2] = (jpeg_c[h, w] - min) * (1.0/(max-
                min))
        return matrix**q
```

Пошук найкращих значень за величиною спектрального компоненту S (рисунок 3.5). Для визначення значення непомітності компонента за власним значенням – S створюється відповідну функції:

0,045455	0	0	0,045455	0,136364	0,272727	0,363636	0,454545
0	0	0,045455	0,090909	0,136364	0,454545	0,454545	0,409091
0,045455	0,045455	0,045455	0,136364	0,272727	0,409091	0,545455	0,409091
0,045455	0,045455	0,090909	0,181818	0,363636	0,681818	0,636364	0,454545
0,090909	0,090909	0,227273	0,409091	0,545455	0,909091	0,863636	0,590909
0,136364	0,227273	0,409091	0,5	0,636364	0,863636	0,954545	0,727273
0,363636	0,5	0,636364	0,681818	0,863636	1	1	0,818182
0,545455	0,727273	0,772727	0,818182	0,909091	0,818182	0,863636	0,818182

Рисунок 3.4 – Показники частотної непомітності компонентів за значенням яскравості

Для підрахування загального значення кожного компонента потрібно буде використовувати багатопоточність, це позитивно відобразиться на часі виконання програми:

```
def rating_s(matrix):
    rating = np.zeros(matrix.shape)
    shf = get_Shf(matrix)
    sf = smart_fix(matrix)
    sf = np.absolute(sf)
    sf = halfSigmoid(sf)
    for i in range(0, 3):
        rating[:, :, i] = shf[:, :, i] * sf[:, :, i]
    return rating

def full_rating_s(ycbcr_image, block=8):
    rating = np.zeros(ycbcr_image.shape)
```

```
def manipulations(h, w, octet):
     nonlocal rating
     octet = octet_dct(octet)
     octet = jpeg_div(octet)
     octet = rating_s(octet)
     rating[(h*len(octet)):((h+1)*len(octet)),
(w*len(octet[0])):((w+1)*len(octet[0]))] = octet
t_matrix = []
for octet in octet_iterator(ycbcr_image, block):
     t = threading.Thread(target=manipulations, args=octet)
     t.start()
     t_matrix.append(t)
for t in t_matrix:
     t.join()
return rating
```

0,045455	0	0	0	0,110359	0,220718	0	0,367864
0	0	0	0	0,110359	0	0,367864	0,331078
0,043898	0,045453	0,036786	0,110359	0	0	0	0
0,043898	0,045193	0,087796	0,147146	0	0	0	0
0	0	0,183932	0,331078	0,441437	0	0	0,478223
0,110359	0	0,331078	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Рисунок 3.6 – Підсумкове значення непомітності компонентів

6. Вирізання самих непомітних блоків блоків

Для того щоб отримати матрицю зображення для подальшого впровадження інформації необхідно розробити функцію до якої на вхід буде подаватися матриця зображення, а на вихід будемо отримувати матрицю після дискретного косинусного перетворення та квантування:

```
def matrix_for_embedding(ycbcr_image, block=8):
     matrix = np.zeros((ycbcr_image.shape[0]//8*8,
     ycbcr_image.shape[1]//8*8, 3))

     def manipulations(h, w, octet):
          nonlocal matrix
          octet = octet_dct(octet)
          octet = jpeg_div(octet)
```

```
            matrix[(h * len(octet)):((h + 1) * len(octet)), (w *
            len(octet[0])):((w + 1) * len(octet[0]))] = octet

      t_matrix = []
      for octet in octet_iterator(ycbcr_image, block):
            t = threading.Thread(target=manipulations, args=octet)
            t.start()
            t_matrix.append(t)
      for t in t_matrix:
            t.join()
      return matrix
```

Для вирізання блоків інформації від зображення створюється функція яка буде це робити багатопоточних. На вхід буде подаватися матриця зображення і інформація про блоків для вирізання, а на виході отримуємо матрицю зображення без цих блоків:

```
def cut_data_to_pict(ycbcr_image, string_data, block=8):
      matrix = matrix_for_embedding(ycbcr_image, block)
      key_iter = i_built_in_place(full_rating_s(ycbcr_image,
      block)).__iter__()
      n_bit = 0
      mas = []
      for i in string_data:
            path_key = key_iter.__next__()
            if i == "1":
                  if round(matrix[path_key[0], path_key[1],
            path_key[2]])%2 == 0:
                  if matrix[path_key[0], path_key[1], path_key[2]] >= 0:
                        matrix[path_key[0], path_key[1], path_key[2]] += 1
                  else:
                        matrix[path_key[0], path_key[1], path_key[2]] -= 1
            elif i == "0":
                  if round(matrix[path_key[0], path_key[1],
            path_key[2]]%2) != 0:
                        if matrix[path_key[0], path_key[1], path_key[2]] >=
                  0:
                              matrix[path_key[0], path_key[1], path_key[2]]
                        -= 1
                        else:
                              matrix[path_key[0], path_key[1], path_key[2]]
                        += 1
                              n_bit+=1
                  mas.append([path_key[0], path_key[1], path_key[2]])

      def manipulations(h, w, octet):
            nonlocal matrix
            octet = jpeg_multiplication(octet)
            octet = octet_idct(octet)
```

```
            matrix[(h * len(octet)):((h + 1) * len(octet)), (w *
        len(octet[0])):((w + 1) * len(octet[0]))] = octet
        t_matrix = []
        for octet in octet_iterator(matrix, block):
            t = threading.Thread(target=manipulations, args=octet)
            t.start()
            t_matrix.append(t)
        for t in t_matrix:
            t.join()
        matrix = np.round(matrix)
        matrix[matrix > 255] = 255
        matrix[matrix < 0] = 0
        return n_bit, matrix.astype(np.uint8)
```

7. Зворотне квантування (помноження компонентів на матриці квантування

Наступною потрібно написати функцію для виконання квантування та зворотного квантування. На вхід методу квантування буде подаватися октет після дискретного косинусного перетворення, та повертатися буде кантований октет, а функція зворотного квантування буде отримувати октет з повідомленням та повертати октет для зворотного квантування:

```
def jpeg_multiplication(matrix):
    jpeg_y, jpeg_c = quantization_tables(90)
    div_oct = np.zeros(matrix.shape)
    for i in range(1, 3):
    div_oct[:, :, i] = matrix[:, :, i] * jpeg_c
    div_oct[:, :, 0] = matrix[:, :, 0] * jpeg_y
    return div_oct
```

Зворотне дискретно-косинусне перетворення.

Далі потрібно написати метод для зворотного дискретного косинусного перетворення. Метод будуть отримувати на вхід октет, та повертати октет з перетвореними значеннями:

```
def octet_idct(octet):
    octet = octet.astype(np.float64)
    dct_oct = np.zeros(octet.shape)

    def idct2(block):
        return idct(idct(block.T, norm='ortho').T, norm='ortho')

    for i in range(0, 3):
        idct_mas = idct2(octet[:, :, i])
        for row in range(8):
            for column in range(8):
```

```
                dct_oct[row][column][i] = idct_mas[row, column]
    return dct_oct
```

Перехід у простір RGB.

Необхідно створити метод зворотний що буде переводити зображення у RGB простір. На вхід буде приходити 3 матриці розміром таким самим як зображення, в кожній матриці будуть значення одного с кольорових просторів, шлях до місця збереження:

```
def ycbcr_to_jpeg(array, path_to_save):
     im = Image.fromarray(array, 'YCbCr')
     im = im.convert('RGB')
     im.save(path_to_save)
```

3.3 Інтерфейс програмного забезпечення

Інтерфейс програмного продукту налічує у собі 6 модулів(Рис.3.7):

1. Поле вибору вхідного зображення.
2. Список операцій: перевірити чи впровадити хеш.
3. Кнопка початку операції.
4. Поле виводу оригінального зображення.
5. Поле виводу хеш-суми.
6. Показник індикатора виконання.

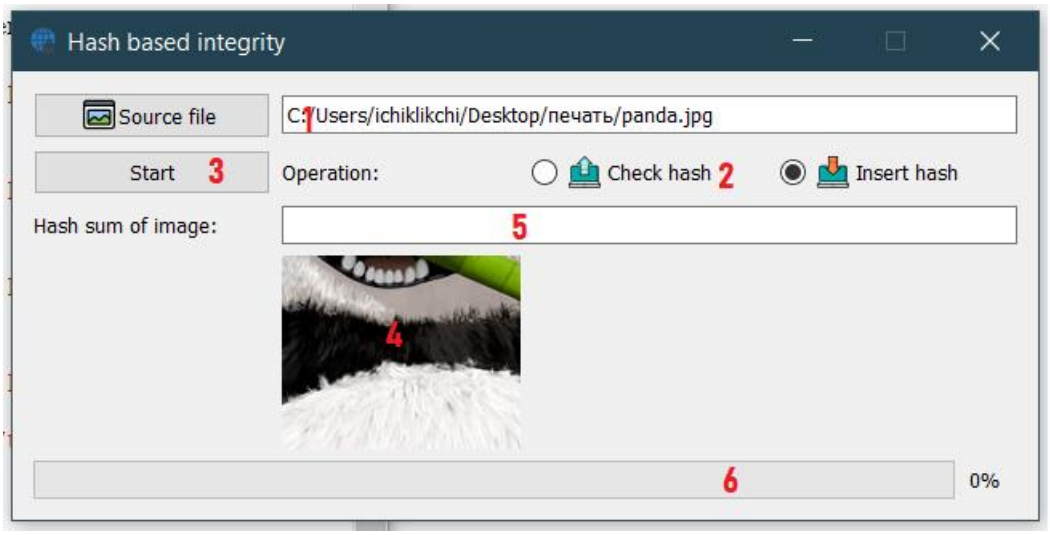

Рисунок 3.7 – Інтерфейс зображення

Детальний опис кожного модуля:

1. Поле вибору вхідного зображення.

У даному полі знаходиться кнопка [Source file] та поле, що виводить шлях до вказаного зображення. Після натиску кнопки з’являється системне вікно файлової мережі.

2. Список операцій: перевірити чи впровадити хеш.

Наступний розділ є списком у якому визначаються дві операції: перевірити хеш та впровадити хеш. Для операції перевірити хеш вхідними параметрами є: путь до зображення яке повинно перевірятися. Для операції впровадження хешу – оригінальне зображення. Операція вважається діючою у випадку вставленої відмітки поряд з її назвою (кружечком).

3. Кнопка початку операції.

Кнопка [Start] знаменує початок операції. За її кліком починається основна робота програми.

4. Поле виводу оригінального зображення.

Поле відображає зменшене оригінальне зображення відразу після натиску кнопки [Start].

5. Поле виводу хеш-суми.

Поле у якому відображається хеш-суми зображення яке було обране.

6. Показник індикатора виконання.

Показує на якому співвідношення програми виконала операцію.

Кінцеві результати програми представлені на рисунках 3.8, 3.9, 3.10.

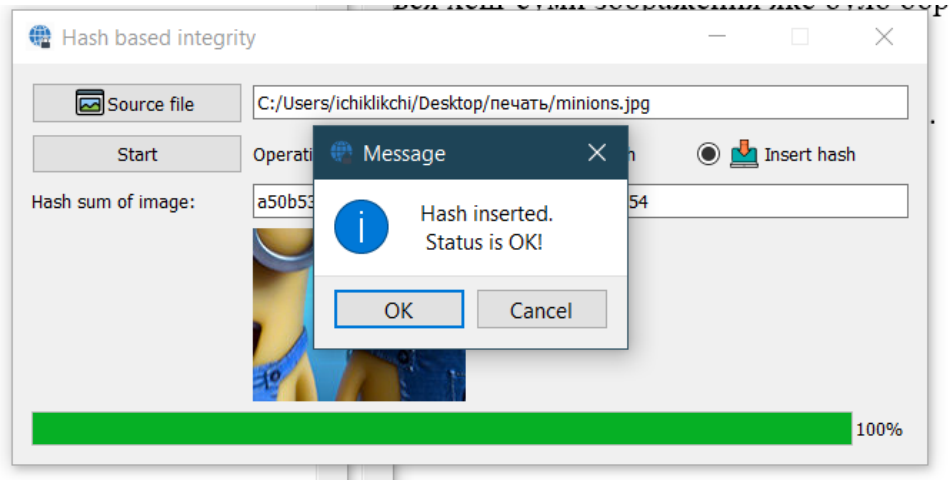

Рисунок 3.8 – Впровадження хеш-суми в зображення

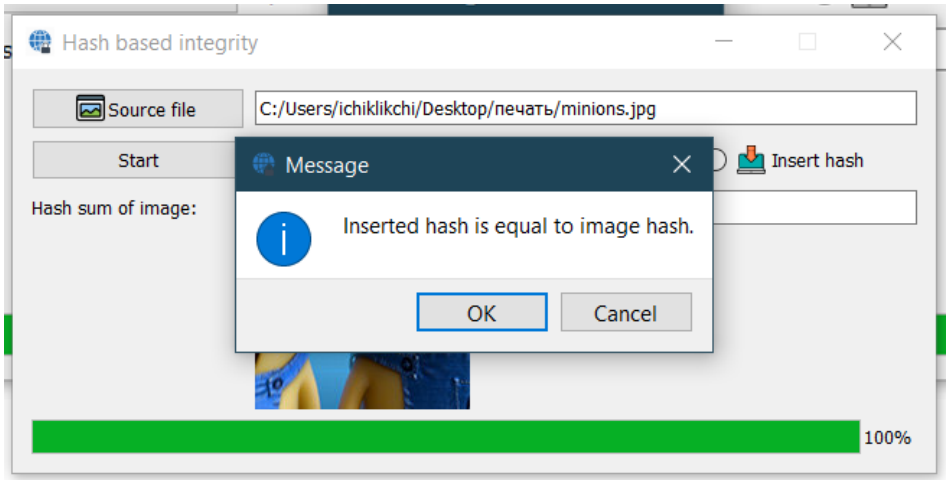

Рисунок 3.9 – Успішне порівняння хеш-суми зображення і вбудованою

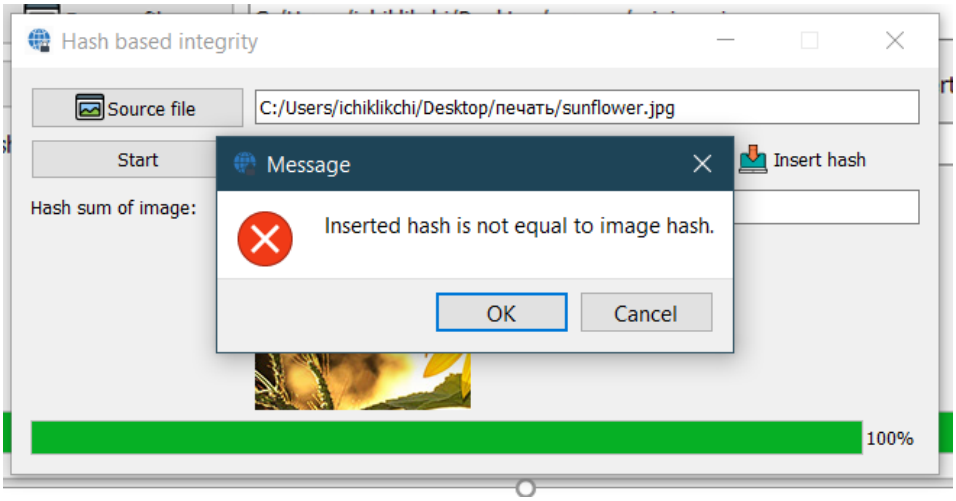

Рисунок 3.10 – Неуспішне порівняння хеш-суми зображення і вбудованою

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

У попередніх розділах дипломного проекту було розглянуто розробку стеганографічного методу, який гарантує цілісність зображення. Ця робота виконується розробником програмного забезпечення у офісному приміщенні. Тому як об'єкт дослідження з охорони праці, вибираємо робоче місце розробника програмного забезпечення.

Офісне приміщення являє собою приміщення загальною площею 20 м^2, і висотою стелі 3 м. У приміщенні знаходиться 4 робочих місця з ПК. Кожне робоче місце обладнане робочим столом площею 1,2 м^2, стільцем та персональним комп'ютером, що складається з монітора, системного блоку, клавіатури та миші.

4.1 Аналіз умов праці і вибір заходів і засобів захисту від небезпечних і шкідливих виробничих факторів

Основним фактором, що безпосередньо впливає на продуктивність праці людей, що працюють з ПК, є комфортні і безпечні умови праці.

Структура робочого місця: стіл: площа – 1,2 м^2, висота – 725 мм; комп’ютерне крісло: висота до сидіння – 660 мм, загальна висота – 820 мм, з механізмом регулювання; системний блок: знаходиться на окремій підставці під столом; монітор: на відстані від 60 до 80 мм до очей оператора, з механізмом регулювання; клавіатура: чорно-біла контрастність; миша: середнього розміру.

Фактори виробничого середовища та трудового процесу:

1. Пил.

Заходи захисту від пилу (ДСТУ-Н Б А 3.2-1:2007):

– вентиляція (місцева і загальнообмінна);
– герметизація джерел пилу разом з аспірацією (місцеве відсмоктування);
– зволоження пилоподібних матеріалів;
– брикетування і гранулювання пилоподібних матеріалів;

– засоби індивідуального захисту – респіратори, протигази, комбінезони, захисні окуляри[18].

2. Шум та вібрації.

Шум погіршує умови праці здійснюючи шкідливу дію на організм людини. Працюючі в умовах тривалої шумової дії випробовують дратівливість, головні болі, запаморочення, зниження пам'яті, підвищену стомлюваність, зниження апетиту, біль у вухах і т. п.

Показник шуму у офісі – 50 дБ(ДСН 3.3.6.037-99). Практичне значення коефіцієнту вище норми[19].

Перелік організаційно-технічних заходів щодо обмеження несприятливого впливу шуму та вібрації на працюючих наведено в ДСН 2.3.6.037-99 та ДСН 3.3.6.039-99, серед яких зменшення шуму та вібрації на шляху розповсюдження засобами ізоляції та поглинання, наприклад, за рахунок використання гумових, поролонових, інших шумо- чи вібропоглинаючих матеріалів, або інших матеріалів аналогічного призначення, що дозволені для оздоблення приміщень органами державного санітарно-епідеміологічного нагляду[20, 21].

3. Мікроклімат у приміщенні.

Мікрокліматичні показники у офісі: у холодний період року: температура повітря 23ºС, відносна вологість 40%, швидкість руху повітря не більш 0,1 м/сек.; у теплий період року: температура повітря 25ºС, відносна вологість 50%, швидкість руху повітря не більш 0,1 м/сек.. Практичні значення в межах норми.

Формовані параметри мікроклімату на робочих місцях повинні бути досягнені, в першу чергу, за рахунок раціонального планування виробничих приміщень і оптимального розміщення в них устаткування з тепло-, холодо- та вологовиділеннями. Для зменшення термічних навантажень на працюючих передбачається максимальна механізація, автоматизація та дистанційне управління технологічними процесами і устаткуванням[22].

4. Важкість праці.

Важкість праці – це характеристика трудового процесу, що відображає переважне навантаження на опорно-руховий апарат і функціональні системи організму (серцево-судинну, дихальну та ін.), Що забезпечують його діяльність. Стереотипні робочі рухи при локальному навантаженні (за участю м'язів кистей та пальців рук) – 40 000 (шкідливі умови – 2 ступінь). Робоча поза – періодичне перебування в зручній позі та/або фіксованій позі до 50% часу зміни; перебування в вимушеній позі від 10 до 25% часу зміни (шкідливі умови – 1 ступінь).

Відповідно до ДСанПіН 3.3.2.007-98 протягом дня мають передбачатися:

– перерви для відпочинку і вживання їжі (обідні перерви);

– перерви для відпочинку і особистих потреб (згідно з трудовими нормами);

– додаткові перерви, що вводяться для окремих професій з урахуванням особливостей трудової діяльності.

5. Напруженість праці.

Напруженість праці – це характеристика трудового процесу, що відображає навантаження переважно на центральну нервову систему, органи чуття, емоційну сферу працівника.

Показники праці для програміста: зміст роботи – рішення складних завдань з вибором за відомим алгоритмом, робота за серією інструкцій (шкідливі умови – 1 ступінь); характер виконуваної роботи – робота за встановленим графіком з можливим його коректуванням у ході діяльності (середнє фізичне навантаження); щільність сигналів (світлових і звукових) і повідомлень в середньому за 1 годину роботи, од. – 170 (середнє фізичне навантаження); розмір об'єкту розрізнення при тривалості спостереження – менше 0,3 мм – 25-50% часу зміни, 0,3-1мм – більше 50% часу (шкідливі умови – 1 ступінь); ступінь відповідальності за результат своєї діяльності – несе відповідальність за виконання окремих елементів завдання (легке фізичне навантаження); кількість елементів необхідних для реалізації простого завдання – більше 10 (легке фізичне навантаження); монотонність виробничої обстановки (час пасивного спостереження за ходом технологічного процесу в% від часу зміни), годину – 70% (легке фізичне навантаження);

наявність регламентоване перерв, та їх тривалість – перерви регламентовані, достатньої тривалості 7% і більше часу зміни (легке фізичне навантаження).

Раціональний, фізіологічно обґрунтований режим праці і відпочинку повинен відповідати таким вимогам:

– запобігати ранньому і надмірному розвиткові втоми працівників;

– сприяти збереженню високої працездатності і оптимального функціонального стану організму працівників протягом зміни;

– забезпечувати високу продуктивність праці;

– сприяти ефективному відновленню фізіологічних функцій під час відпочинку.

6. Забарвлення і коефіцієнти відображення.

Забарвлення приміщень і меблів повинна сприяти створенню сприятливих умов для зорового сприйняття.

Коефіцієнт відображення у офісі: для селі – 60%, для стін – 40%, для підлоги – 20%, для робочої поверхні – 20%. Нормативний коефіцієнт відображення (ДСТУ 7239:2011): для селі – 60-70%, для стін – 40-50%, для підлоги – 20%, для робочої поверхні – 20-40%. Практичне значення коефіцієнту відображення відповідає нормі.

Відображення, включаючи відображення від вторинних джерел світла, повинне бути зведено до мінімуму. Для захисту від надмірної яскравості вікон можуть бути застосовані штори і екрани. У приміщеннях, де знаходиться комп'ютер, необхідно забезпечити нормативні величини коефіцієнта відбиття.

7. Освітлення.

Неправильний напрямок світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого.

Освітлення у офісі(робота середньої точності): коефіцієнт природнього освітлення – 1,2 % , загальна освітленість – 225 лк, комбінована освітленість – 366 лк. Нормативне значення освітленості(ДСТУ 7239:2011): коефіцієнт природнього освітлення – не нижче 1 % , загальна освітленість – 200 лк, комбінована

освітленість – 300 лк. Приміщення відзначається підвищеною освітленістю робочої зони.

Коефіцієнт пульсації не повинен перевищувати 5% (ДСанПіН 3.3.2.007-98). Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300–500 лк. Світильники місцевого освітлення слід встановлювати таким чином, щоб не створювати відблисків на поверхні екрана, а освітленість екрана має не перевищувати 300 лк [23].

4.2 Аналіз техногенних небезпек і вибір заходів і засобів забезпечення безпеки у надзвичайних ситуацій

Техногенні надзвичайні ситуації виникають у результаті раптового виходу з ладу машин, механізмів та агрегатів, що супроводжується значними порушеннями виробничого процесу, вибухами, утворенням осередків пожеж, радіоактивним, хімічним чи біологічним зараженням місцевості, які призвели чи можуть призвести до значних матеріальних втрат та враження чи загибелі людей. Основним наслідком аварій та катастроф є пожежі на виробництві.

Пожежа – явище небезпечне і неконтрольоване. Здатне принести масу шкоди здоров'ю і майну.

Основні причини пожеж у офісних приміщеннях:

– короткі замикання;

– використання несправного електрообладнання;

– застосування обігрівальних приладів відкритого типу;

– куріння в недозволених місцях;

– неправильне поводження з обладнанням або небезпечними речовинами;

– цілеспрямовані підпали.

Засоби пожежної безпеки:

1. Пожежні сповіщувачі.

Димові сповіщувачі дозволяють виявити певну концентрацію частинок диму в повітрі. Теплові сповіщувачі реагують на збільшення температури. Сповіщувачі полум'я реєструють сплески ультрафіолетового та інфрачервоного випромінювання, якими характеризується вогонь.

2. Вогнегасники.

Пожежна безпека забезпечується наявністю вогнегасників, порошкових або вуглекислотних.

Для офісних приміщень найкращим є вогнегасники з порошком ПФ: склад за основним компонентом – фосфорно-амонійні солі з добавками; вологість – не більше 0,5; насипна маса – 0,8 – 0,9 г/смз; застосування – гасіння газів, рідин, що розлилися, електроустановок під напругою та деревини. Для офісного приміщення з чотирма ПК – необхідно два вогнегасника – один ВП-5.

3. Нагляд за пожежною безпекою

Згідно з вимогами, в офісі необхідно вести журнал реєстрації протипожежного інструктажу, інструкцію про заходи пожежної безпеки та інструкції про порядок дій персоналу при пожежі. Співробітник, відповідальний за пожежну безпеку, повинен мати атестацію навчального центру МНС.

Визначення категорії приміщення.

Методика визначення категорій приміщень та будівель за вибухопожежною та пожежною небезпекою регламентується НАПБ Б.03.002-2007[24].

Приміщення, у якому знаходиться робоче місце працівника можна віднести до категорії В – вибухо-пожежоне-безпечна. До цієї категорії відносять приміщення, в яких присутні характеризується наявністю горючих матеріалів і речовин, здатних тільки горіти.

Основним профілактичним заходом щодо попередження пожеж і вибухів від електрообладнання є правильний вибір та експлуатація такого обладнання у вибухо- та пожежонебезпечних приміщеннях.

4.3 Біоритми – фізіологічні основи запобігання небезпеці, прояву ризику

Біоритми будуть розраховуватися для інженера-програміста з подальшими датами:

Дата народження 19.08.1998.

Дата дослідження 25.11.2020.

1. Кількість повністю прожитих років: *H=2020 -1998 -1=21*
2. Кількість високосних років серед повністю прожитих: *L=5*
3. Кількість прожитих днів у рік народження: *T=134*
4. Кількість прожитих днів у поточному році до заданої дати: *R=330*
5. Загальна кількість прожитих днів: *D=8134*
6. Кількість повних прожитих циклів: *F=353*
7. День фізичного циклу залишок: *F* =179*
8. Для емоційного: *E=290*
9. Для емоційного циклу залишок: *E*=219*
10. Для інтелектуального: *I=246*
11. Для інтелектуального циклу залишок: *I*=260*

Виходячи з розрахованих результатів створюємо таблицю результатів і діаграма біоритмів:

Таблиця 4.1 - Біоритми

Досліджуваний біоритм	Дата дослідження	Дата народження	Кількість прожитих днів	Кількість повних періодів біоритмів	Залишок днів	Дата проходження крізь 0	Фаза МБР
Фізичний Емоційний Інтелектуальний	25.11.2020	19.08.1998	*8134* *8134* *8134*	*353* *290* *246*	*179* *219* *260*	23.05.2021 02.07.2021 12.08.2021	Позитивна Позитивна Позитивна

Діаграма біоритмів (Рис 4.1):

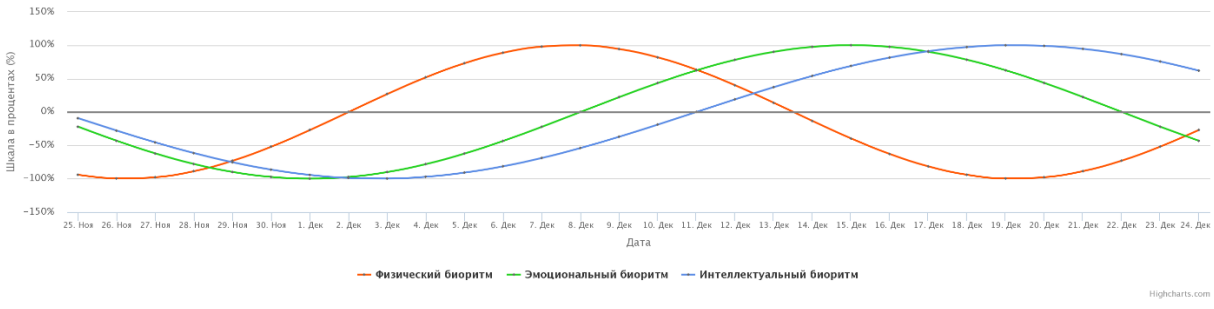

Рисунок 4.1 - Діаграма біоритмів

ВИСНОВКИ

Результатом кваліфікаційної розроблений метод контролю цілісності зображень, що ґрунтуються на використанні цифрових водяних знаків, розробка рейтингу компонентів зображення, методом оцінки кожного компоненту на предмет того, наскільки помітне буде впровадження в нього згідно психовізуальної моделі JPEG, та впровадження в вибрані компоненти, починаючи з самих непомітних, а також програмне забезпечення яке реалізує цей метод.

В першу чергу було вибрано та проаналізовано методи контролю цілісності зображень. За результатами аналізу обрано метод який став основою нового методу.

Наступним кроком було досліджено хеш-функцію SHA-1 та алгоритм стиснення зображення JPEG. За результатами дослідження було виявлено основні етапи стиснення інформації та вибрано етап після квантування блоків зображення як етап для вирізання блоків.

Створено метод, за яким відбувається впровадження хеш-суму у найменш помітні місця.

Реалізовано метод у програмному середовищі Python. Програма має візуальний інтерфейс та працює в операційній системі Windows, Linux.

Дослідження та розробка у кваліфікаційні роботі у подальшому можуть бути використані для розробки контролю цілісності у структуру алгоритму JPEG.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Климов, Е. Механізми контролю цілісності даних [Електронний ресурс] / Є. Климов. - Режим доступу: URL: http://www.iso27000.ru/chitalnyi-zai/kriptografiya/ mehanizmy-kontrolya-celostnosti-dannyh. -2012. (дата звернення: 25.07.19)
2. Михайличенко О.В., Коробейников А.Г, Каменєва С.Ю. Застосування стеганографічних методів приховування інформації в нерухомих зображеннях // Праці міжнародних науково-технічних конференцій «Інтелектуальні системи» (IEEE AIS'06) і «Інтелектуальні САПР» (CAD-2006) ": в 3 т. М .: Фізматліт, 2006. Т .2. - С.511 -515.
3. О. В. Генне, ООО "Конфидент" Опубликовано: журнал "Защита информации. Конфидент", №3, 2010
4. Підсистема контролю цілісності [Електронний ресурс] / ЗАТ "Астра-СТ". - Режим доступу: URL: http://mirage.astra-st.ru/rus/main.html. - 2004. (дата звернення: 28.20.20)
5. Хорошко В.О., Азаров О.Д., Шелест М.Е. Основи комп'ютерної стеганографії: уч. посібник для студентів і аспірантів / - Вінниця: ВДТУ, 2003.
6. SHA-1. Режим доступу: URL: https://en.wikipedia.org/wiki/SHA-1 – 2010 (дата звернення: 28.20.20)
7. JPEG. URL: https://ru.wikipedia.org/wiki/JPEG.
8. Petitcolas, F.A.P, Anderson R. J., Kuhn M. G. Attacks on Copyright Marking Systems. *Second workshop on information hiding*. 2008. Vol. 1525, No 7. P. 218-238.
9. Коробейников А.Г., Прохожев М.М., Михайличенко О.В., Хоанг З. Вибір коефіцієнтів матриці дискретно-косинусного перетворення при побудові стеганографічних систем. *Вісник комп'ютерних та інформаційних технологій*. 2008. 2 листоп. (№ 11). С. 12-17.

10. Loeffler C., Ligtenberg A., Moschytz G. Practical Fast 1-D DCT Algorithms with 11 Multiplications. *Proc. Int'l. Conf. on Acoustics, Speech, and Signal Processing*. 2009. Vol. 15, No 1. P. 988-991.
11. Steganography: Implications for the Prosecutor and Computer Forensics Examiner. URL: http://www.fbi.gov/hq/lab/fsc/backissu/july2004/researc/ 2004 03 research01.htm.
12. Celik M. U., Sharma G., Tekalp A. Image Steganalysis Using Rate-Distortion curves. *SPIE: Security, Steganography, and Watermarking of Multimedia Contents.* 2004. Vol. 53, No 4. P. 19-22.
13. Кірмічієва А.С., Кушніренко Н.І., Яковенко А.А., Калашников Н.В., Лозан А.Е. Метод впровадження інформації в цифрові зображення JPEG, що мінімізує психовізуальні спотворення для малих обсягів впроваджуваної інформації. *Інформатика та математичні методи в моделюванні.* 2018. Т. 8 (№ 4). С. 313-323.
14. Python. URL: https://www.python.org/.
15. PyQt. URL: https://ru.wikipedia.org/wiki/PyQt.
16. NumPy. URL: https://ru.wikipedia.org/wiki/NumPy.
17. Python_Imaging_Library.URL: https://ru.wikipedia.org/wiki/Python_Imaging_Library.
18. ГОСТ 12.1.007-76. Система стандартів безпеки праці (ССБП). Шкідливі речовини. Класифікація і загальні вимоги безпеки. [Чинний від 1977-01-01]. Вид. офіц. Москва, 1977. 10 с. (Інформація та документація).
19. ГОСТ 12.0.003-2015. Система стандартів безпеки праці (ССБП). Небезпечні і шкідливі виробничі фактори. Класифікація. [Чинний від 2017-03-01]. Вид. офіц. Москва, 2017. 20 с. (Інформація та документація).
20. ДСН 3.3.6.037-99. Санітарні норми виробничого шуму, ультразвуку та інфразвуку. [Чинний від 1999-12-01]. Вид. офіц. Київ, 1999. 17 с. (Інформація та документація).

21. ДСН 3.3.6.039-99. Державні санітарні норми виробничої загальної та локальної вібрації. [Чинний від 1999-12-01]. Вид. офіц. Київ, 1999. 13 с. (Інформація та документація).
22. ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень. [Чинний від 1999-12-01]. Вид. офіц. Київ, 1999. 12 с. (Інформація та документація).
23. ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. [Чинний від 1998-12-10]. Вид. офіц. Київ, 1999. 10 с. (Інформація та документація).