МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 681.3

О.В. Тимченко
Національний університет “Львівська політехніка”

СТЕГАНОГРАФІЧНІ МЕТОДИ ВПРОВАДЖЕННЯ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ У ЦИФРОВІ ЗОБРАЖЕННЯ ФОРМАТУ BMP

Докладно розглянуто методи впровадження цифрових водяних знаків для захисту авторських прав на файли зображень формату BMP.

The methods of defence of digital thread-marks are thoroughly considered for defence of copyrights files of images of the BMP format.

Вступ

Практичного значення і широкого застосування набуває вміщення неявних даних як метод захисту авторських прав, а саме вміщення в цьому разі є впровадженням цифрових водяних знаків (ЦВЗ) за допомогою стеганографічних методів. Цифрові водяні знаки можуть бути вбудовані в DVD, цифрові зображення, відео, музику.

Розглянуті в [1, 2] методи формування цифрових зображень показують, що графічні файли можуть з успіхом використовуватись для приховування додаткової таємної інформації завдяки значній надлишковості, яку практично неможливо усунути із реальних зображень.

Для визначення ефективності впровадження ЦВЗ порівняємо найчастіше застосовувані техніки приховування даних у графічних файлах.

1. Загальна модель стеганографії

На рис. 1 показано загальну модель вміщення ЦВЗ. Різноманітні методи відрізняються алгоритмами вміщення даних, проте загальна схема однакова для всіх методів. У загальній моделі стеганографії користувач вміщує неявні дані через застосування, по-перше, деякого перетворення, а далі підміни низки незначущих бітів. Незначущі тут означає, що певна випадково вибрана і достатньо велика підмножина всього зображення може бути змінена так, щоб зміна була непомітною для стороннього спостерігача [1].

Основними вимогами стеганографії є відсутність можливості за допомогою візуального сприйняття викривання вміщеної інформації або можливості її відшукання за допомогою відповідного алгоритму, без необхідності відтворення оригінального зображення.

В загальному випадку можна побудувати функціонуючу стегосистему без додаткових даних – ключів стенографічного алгоритму.

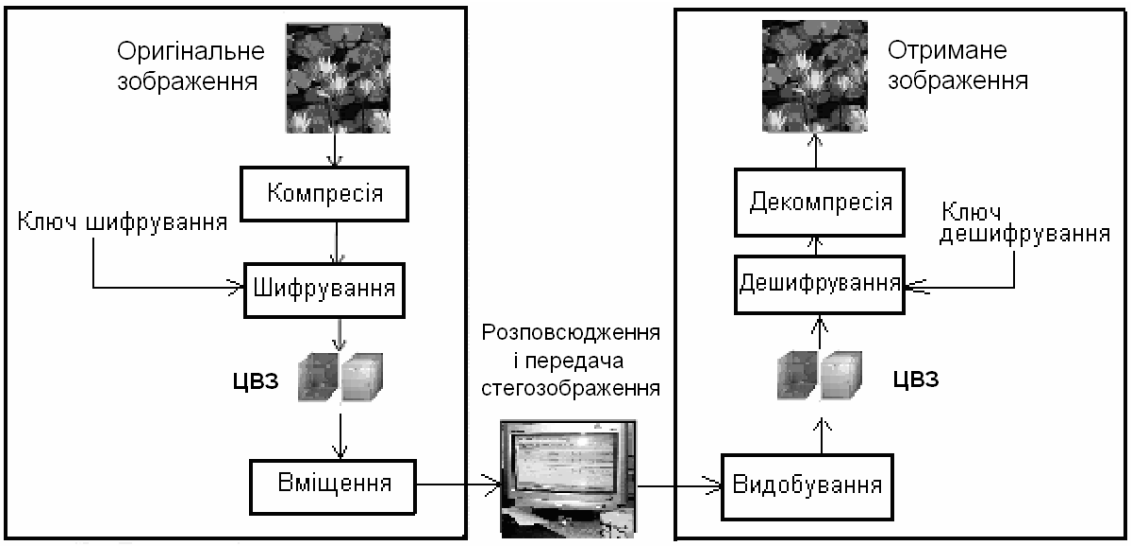


Рис. 1. Модель вміщення ЦВЗ

Визначення 1. Набір $\sum=<C,M,D,E>$, де C – множина можливих контейнерів; M – множина явних відомостей, $|C| \geq |M|$; $E: C\times M\rightarrow C$ і $D: C\rightarrow M$ – функції вміщення і видобування відомостей, прихованих в контейнерах, причому $D(E(c, m))=m$ для всіх $m\in M$ і $c\in C$ назвемо стегосистемою без ключа.

З визначення 1 випливає, що безпечність систем без ключа полягає в утаємненні використаних стеганографічних перетворень E і D. Проте це завжди є основною вадою систем охорони інформації. Якщо відомо, що наш противник знає алгоритми E і D, використані для приховування ЦВЗ, то треба пам'ятати, що він легко відшукає і змінить інформацію з прийнятих стегограм.

Визначення 2. Стегосистемою з таємним ключем (симетричним) назвемо набір $\sum=<K,C,M,D,E>$, де C – множина можливих контейнерів; M – множина явних відомостей, K – множина стегоключів; $E_k: C\times M\times K\rightarrow C$ і $D_k: C\times K\rightarrow M$ – стеганографічне перетворення, таке, що $D_k(E_k(c,m,k),k)= m$ для кожного $m \in M, c \in C$ і $k \in K$.

Такий тип стегосистеми вимагає існування безпечного каналу для обміну ключами. В симетричних алгоритмах ключ шифрування визначається з ключа дешифрування і навпаки. Тобто ключ шифрування і дешифрування є тим самим.

Системи з явним ключем є асиметричними або приватними стегосистемами. Вони спроектовані так, що ключ для шифрування відрізняється від ключа, що застосовується для дешифрування.

Визначення 3. Стегосистемою з явним ключем (асиметричним) назвемо набір $\sum=<K,C,M,D,E>$, де K – множина пар стегоключів (ключ явний k_1 використовується для шифрування, ключ таємний k_2 використовується для дешифрування); $E_k: C\times M\times k_1\rightarrow C$ і $D_k: C\times k_2\rightarrow M$ – стеганографічне перетворення, таке, що $D_k(E_k(c,m,k),k)=m$ для кожного $m\in M, c\in C$ і $k_n\in K$.

Знаючи ключ k системи, можна простими засобами використати алгоритми E_k і D_k. Якщо один з користувачів стегосистеми хоче переслати до іншого користувача відомість $m\in M$, то застосовує шифр E, утворюючи контейнер $c = E(m)$, який пересилається телекомунікаційним каналом. Отримувач дешифрує c за допомогою D і одержує $D(c) = m$.

Таке розв'язання з двома ключами ліквідує багато проблем симетричних алгоритмів. Тільки ключ дешифрування (приватний) треба охороняти. Ключі шифрування (приватний) роздають кожній особі, яка хоче з нами зв'язатися – особа без особливої підготовки може вживати ключ шифрування для передавання відомості, проте прочитати цю відомість може лише той, хто має ключ дешифрування. У разі такого розв'язання публічний ключ можна помістити на візитці або на стороні WWW.

2. Метод LSB (Least Significant Bit – найменших значущих бітів) [3]

З погляду стеганографії найвигіднішими є зображення в форматі BMP. Проста будова та великий обсяг дають змогу вільно модифікувати їхню величину, без необхідності декомпресії, та без наражання прихованої відомості на пошкодження, що виникає при подальшій компресії файла.

Метод LSB – це основна і найпростіша техніка для вбудовування даних в файлах без таблиці кольорів, таких, як, наприклад, BMP або GIF. Байт поданий бінарно за допомогою 8 величин з множини {0,1} з відповідною вагою – 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0.

Найменший значущий біт має вагу 2^0. Оскільки зображення записане в 24-бітовій глибині кольору, де кожний піксел подається 3 байтами, кожний з яких відображає величину однієї з складових кольору, то один піксел може мати 2^{24} =16777216 кольорів. Якщо наймолодший біт буде змінений – його величина буде замінена іншою величиною з множини {0,1}, одночасно відтінок кольору зміниться. Проте зміна ця буде наскільки малою, що око не в стані зауважити різницю між модифікованим та оригінальним кольором.

Метод LSB дає змогу приховати велику кількість даних *Q* навіть у невеликому зображенні:

$$Q = (X \cdot Y) \cdot (K/8)/8,$$

де *X, Y* – ширина і висота зображення, пікселі; *K* – глибина кольору, біти.

Приклад вміщення одного байта тексту в восьмибітовому зображенні:

дані є фрагментом зображення в бінарному поданні:

00101101 11010110 00101101 01101001 11010110 10000101 01110100 11001010

а знак для вміщення – K, що в бінарному поданні має вигляд 10010111.

Тоді той самий фрагмент зображення після вміщення знака набуде вигляд:

00101101 11010110 00101100 01101001 11010110 10000101 01110101 11001011

Як бачимо з наведеного прикладу, знак “K” прихований у восьми бітах зображення. Видно також, що не всі наймолодші біти поданого фрагмента зображення були змінені. Лише три з них були змінені, що показує – виконані зміни зображення мінімізовані.

Стосовно методу LSB існує низка проблем. Якщо вміщувана таємна відомість значно коротша від величини оригінального зображення, це може призвести до декількох вад:

- вміщуючи ЦВЗ тільки в перших секціях оригінального зображення – починаючи з першого байта, яким подано зображення (тобто минаючи заголовок файла), можна змінити статистичні властивості зображення. Це призводить до того, що блок, в який вміщено ЦВЗ, буде мати інші статистичні властивості, ніж решта зображення;
- вибираючи ЦВЗ, не можна розпізнати його кінець. Крім ЦВЗ, отримуємо велику кількість нічого не значущих даних.

3. Метод обчислення різниці величини пікселів (PVD- Pixel Value Differencing) [4]

Метод LSB доволі популярний і простий алгоритмічно, проте чим більше бітів має ЦВЗ, тим більш спотворене буде стегозображення. Очевидно, не кожний піксел може бути модифікований такою самою мірою, без того, щоб не викликати видимих змін зображення. Зміни в однорідних областях, що характеризуються невеликими різницями у величині пікселів, стають більш видимі, ніж в областях з сильним коливанням відтінків. Пікселі, які розміщені на краях областей з різною інтенсивністю, можуть містити більше інформації. Метод PVD використовує цю властивість, поміщаючи великі порції прихованих даних в зображення з великим контрастом.

Розглянемо цей метод для цифрового зображення з 8-бітовою шкалою сірого. Метод ділить зображення на ненакладані двопіксельні блоки. Поділ пробігає рядками по лінії зигзагу, можна також застосувати псевдовипадковий вибір блоків (рис.2).

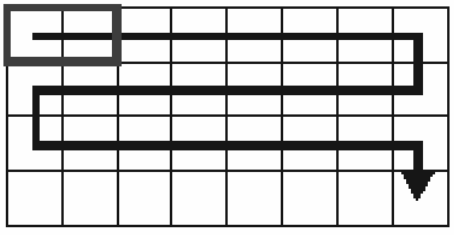


Рис.2. Поділ і пробігання зображення

Кожний з блоків зараховуємо до певної категорії з огляду на різницю величини складових точок. Залежно від величини цієї різниці у такому блоці кодується відповідна кількість бітів прихованої інформації.

Опис методу. Для кожного блока з двох пікселів g_i та g_{i+1} обчислюємо різницю їхніх значень

$$d = g_{i+1} - g_i.$$

Очевидно, що $d \in < -255, 255 >$. Величина різниці близько 0 показує, що піксел розміщений в однорідній області, різниця близько 255 означає великий контраст між пікселами. З погляду симетрії різниці класифікуємо її на основі величини до однієї з меж R_i, де i=1,2…,m. Верхню і нижню величини діапазону позначимо через l_i та u_i. Сума меж R_i ділить діапазон <0,255> на n підмеж так, що

$$\bigcup_i R_i = \langle 0, 255 \rangle \quad \text{та } l_1=0,\ u_m=255,\ l_{i+1}=u_i+1.$$

Ширина меж w_i обчислюється як $w_i=u_i-l_i+1$ і повинна бути ступенем 2. Ширина меж, до яких відноситься ця різниця, дає відповідну кількість бітів, які можна закодувати у цьому блоці. Ширина меж R_i вибирається на основі характеристик сприйняття зору і є меншою для невеликих різниць величин пікселів і більшою для сильних різниць. Прикладом величини ширини меж для 0, 8, 16, 32, 64, 128 які ділять межі <0,255>, є такі підмежі: <0,7>, <8,15>, <16,31>, <32,63>, <64,127>, <128,255>. Від вибраних величин залежить продуктивність (виміряна як кількість бітів, можливих для приховування) або якість методу. Різниця, що належить до меж R_k , має індекс k. Всі величини різниць, що мають однаковий індекс, "порівняно близькі". Якщо ця різниця буде замінена іншою величиною різниці з тим самим індексом, то зміна буде незначною. Тому запропонований метод кодує визначену послідовність бітів у блоках двох пікселів, модифікуючи величину різниці між ними.

Кодування. Подамо приховану інформацію як послідовність бітів. Вибираємо два послідовні пікселі із зображення. Кількість бітів, яка можливо закодувати, залежить від ширини межі R_k, до якої належить різниця величин d і обчислена як

$$n = \log_2(w_k) = \log_2(u_k - l_k + 1).$$

З прихованої інформації вибираємо підпослідовність з n бітів. Її величину позначимо b. Нова величина d' різниці:

$$d' = \begin{cases} l_k + b, d \geq 0; \\ -(l_k + b), d < 0. \end{cases}$$

Оскільки величина $b \in <0, u_k - l_k>$ то нова величина d' має такий самий індекс k, що і оригінальна різниця d. Тому вони "порівняно близькі". Зміна значення різниці потім сприймається як незначна. Кодування величини b відбувається через зворотне обчислення величин пікселів (g_i', g_{i+1}') на основі різниці d'. Нова величина піксела визначається за допомогою функції $f((g_i, g_{i+1}), m)$, записаної як:

$$f((g_i, g_{i+1}), m) = (g_i', g_{i+1}') = \begin{cases} \left(g_i - \left\lceil \frac{m}{2} \right\rceil, g_{i+1} + \left\lfloor \frac{m}{2} \right\rfloor \right), d \bmod 2 = 1; \\ \left(g_i - \left\lfloor \frac{m}{2} \right\rfloor, g_{i+1} + \left\lceil \frac{m}{2} \right\rceil \right), d \bmod 2 = 0. \end{cases}$$

де $m=d'-d$.

Верхня частина функції відповідає випадку, коли $g_{i+1}-g_i=d'$. Зміна значень розкладається рівномірно на обидва пікселі.

У випадку, коли настає перевищення області можливих значень для окремого піксела, нова величина не буде належати до меж <0,255>. Для вирішення цієї проблеми необхідно упевнитись в тому, чи піксел у цьому блоці може потенціально перевищити дозволену величину, щоб пропустити їх під час приховування інформації. Застосування цієї самої процедури в процесі видобування даних дає змогу визначити, чи цей блок був використаний для кодування інформації. Процедура полягає в обчисленні пари значень $(\hat{g}_i, \hat{g}_{i+1})$ на основі зворотного обчислення функції величини

піксела $f((g_i,g_{i+1}), u_k\text{-}d)$. Оскільки u_k є верхньою межею R_k, пара $(\hat{g}_i, \hat{g}_{i+1})$ буде характеризувати найможливішу різницю величин. Якщо якась з отриманих величин перевищує дозволене значення, то цей піксел може потенційно перевищувати задану величину і завжди обминається під час кодування. Та сама процедура використовується під час відшукування закодованої інформації з метою упевнення, чи блок містить необхідні дані.

Приклад вміщення інформації за методом PVD подано нижче на рис. 3.

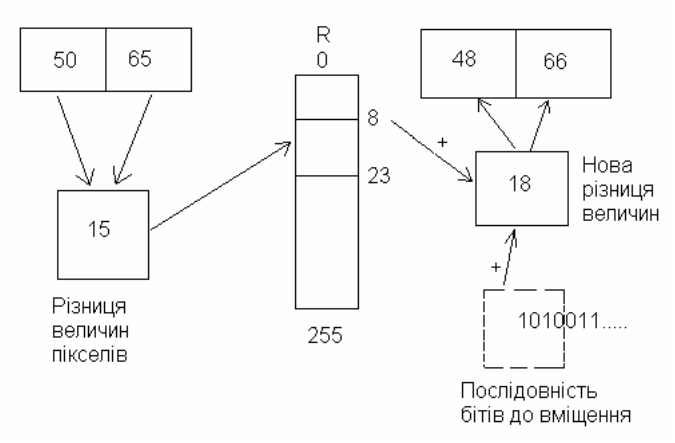


Рис. 3. Приклад вміщення інформації за методом PVD

Рисунок ілюструє кодування інформації в блоці з двох пікселів. Початкові величини пікселів (50,65) дають різницю d=15. Величина ця належить до меж R_k=<8,23>.

Ширина меж w_k=23–8+1=16=2^4. Кількість бітів, котрі можна помістити в блок n=$\log_2(w_k)$=$\log_2(2^4)$=4. З послідовності бітів ЦВЗ вибираємо чотири послідовні біти. Величина створеної підпослідовності – 10. Нова величина різниці значень пікселів d'=l_k+b=8+10=18. На основі зворотної функції обчислюємо нові значення пікселів у блоці (48, 66).

4. Стеганографія з використанням стохастичної модуляції [4]

Метод з використанням стохастичної модуляції вміщує біти відомості в пікселах зображення через додавання до зображення шуму з розподілом, симетричним щодо нуля. Носієм для методу є цифрове зображення з 8-бітовою шкалою сірого. Зауважимо, що якщо $\{s_i\}$ має нормальний розподіл N(0,σ) і якщо z_i є випадкова змінна, рівномірно розподілена в діапазоні {–1, 1}, то $\{z_i s_i\}$ має розподіл Гаусса N(0,σ). Іншими словами, розподіл Гаусса з випадково змінними знаками залишається гауссівським.

Приймемо, що відомість m_i – випадкова послідовність {-1,1}, m_i має середнє значення 0. Визначимо функцію парності P для величини пікселів $P(x,s)\in$ {-1,1}, для $x\in$ {0,…,255} і s>0, де s є цілим параметром, при $P(x,s)$=0 для s=0.

Функція, визначена на величині точки зображення, буде визначати біти, вміщені у відомості. Функція повинна мати таку властивість асиметрії для кожного x:

$$P(x+s,s)=-P(x\text{-}s,s) \qquad \text{для } s\neq 0.$$

Наприклад, для s=1 можна прийняти $P(x,1)$ для x=0,1,2,…або $P(x,1)$=1, 1, –1,–1,1, 1, –1, –1,… Загальною умовою для s>0 є те, що перший сегмент 2s величини може бути будь-яким, проте кожний наступний сегмент 2s величини мусить бути протилежним до попереднього. Прикладом функції парності може бути:

$$P(x,s)=(-1)^{x+s}, \quad x\in [1,2s].$$

Необхідно зауважити, що функція парності P, крім величини, x залежить від величини параметра s. Це істотно, тому що в іншому випадку неможливим було б знаходження функції, яка відповідає заданій вище умові асиметричності для всіх величин x і всіх s.

Вміщення інформації. Маючи вибрану функцію парності, можна розпочати вміщення таємної відомості. Вміщення може відбуватися згідно з заданою послідовністю або в псевдовипадковій послідовності, згенерованій на основі певного ключа. Генератор псевдовипадкових чисел повинен генерувати числа з щільністю такою самою, як шум, накладений на зображення під час вміщення. Згенерований генератором шум назвемо стегошумом. Для кожного піксела вздовж лінії вміщення генеруємо один відлік стегошуму, заокругленого до цілого числа s. Якщо s=0, величина піксела x не модифікована і переходимо до наступного піксела. Якщо $s \neq 0$, впевнюємось, чи $P(x+s, s) = m$, де m є бітом відомості для вміщення. Якщо це виконується, то модифікуємо x до $x+s$ і переходимо до наступного піксела. Якщо виконується протилежна умова чи $P(x+s, x) = -m$, модифікуємо x до x-s. Позначивши величину піксела в стегозображенні після модифікації як x', можемо операцію вміщення записати як:

$$x_i' = x_i + m_i P(x_i + s_i, s_i)\, s_i$$

Можна сказати, що замість того, щоб додавати сигнал $\{m_i s_i\}$ до зображення, ми додаємо $\{v_i s_i\}$, де $v_i = m_i P(x_i + s_i, s_i)$. Згідно з умовою біти відомості утворюють випадкову послідовність 1 і –1. Оскільки зображення і стегошум незалежні від вміщеного секрету, змінна v_i також псевдовипадкова послідовність величин 1 і –1. Для цього v_i має таку саму характеристику розподілу, як стегошум. Ускладненням в методі є граничні величини пікселів 0 та 255. Тоді амплітуда шуму, що додається до зображення, повинна бути скорегована. Якщо $x_i + s_i > 255$, величина x' вибирається як найбільша величина, що менша від або дорівнює 255 з заданою парністю m_i. Аналогічно для $x_i - s_i < 0$ величини x' вибирається як найменші з значень, що більші за або дорівнюють 0 з парністю m_i.

Видобування вміщеної відомості достатньо просте. Генеруємо ту саму псевдовипадкову послідовність, визначаючи послідовність бітів, в яких вміщена інформація. Для кожної наступної величини піксела визначаємо величину функції парності вигляду:

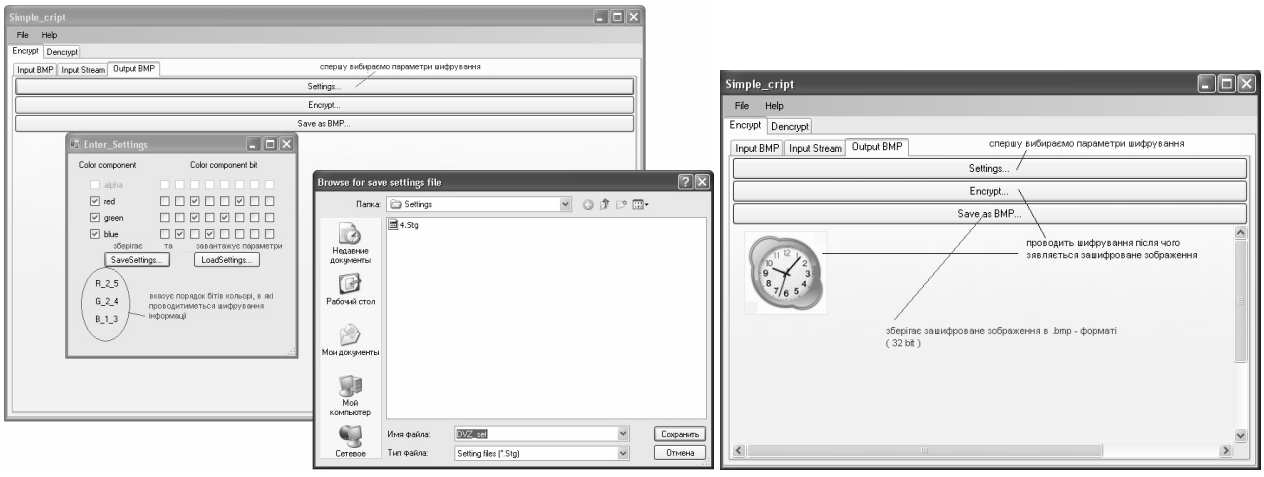
$$m_i = P(x_i, s_i).$$

Ненульова величина функції парності утворює необхідну видобувану відомість. Треба зауважити, що той самий ключ, вжитий для визначення послідовності, вміщеної в пікселах, використовується для визначення послідовності s_i.

Приклад впровадження ЦВЗ

Розглянуті методи формування цифрових зображень реалізовані у програмі впровадження ЦВЗ **Simple_cript** для формату .bmp. Програма дає змогу вбудовувати зображення у будь-які біти графічного файла за допомогою закладки **setting**, яка в цьому разі є ключем. Окрім того, програма дає змогу візуалізувати ЦВЗ у випадку, якщо це один з графічних форматів, або записати зображення ЦВЗ.

Вибір параметрів впровадження фактично є ключем, без знання якого виявити ЦВЗ практично неможливо. На рис. 4, а, б показано перегляд ЦВЗ та вибір місць його розміщення.



а б

Рис. 4. Вибір розміщення ЦВЗ (а) і його візуалізація (б)

Для запису ЦВЗ на рис. 4, а вибрані біти: R_2_5 (червоний), G_2_4 (зелений), B_1_3 (синій).

Фактично розміщення бітів є ключем, що зберігається як файл DVZ_set.txt.

Результат впровадження очевидний – внаслідок невдалого вибору ключа спотворення проявляються в усіх кольорах (рис.5).

Рис. 5. Приклад невдалого впровадження ЦВЗ – ліва частина зображення має значні спотворення

У разі впровадження ЦВЗ лише у молодші біти зображення, що захищають спотворення, на око не помітні – початковий і вихідний файли “Водяные лилии.jpg” та “Водяные лилии.bmp” практично ідентичні.

Висновки

Існуючі формати графічних файлів задають різну вагу колірним компонентам, коефіцієнтам перетворень тощо через позиційне кодування складових зображення, що дає змогу модифікувати велику частину бітів зображення з метою приховування ЦВЗ без візуального погіршення їхнього сприйняття.

Розглянуті методи вміщення ЦВЗ на основі загальної моделі стеганографії у цифрові зображення формату BMP показують їхню високу ефективність і значний обсяг для можливого впроваджування даних.

1. Тимченко О.В. Техніка стеганографії як метод передачі даних в цифрових зображеннях // Зб. наук. пр. ІПМЕ НАН України. – Вип.33. – К., 2006. – С.180–193. 2. Nettavali A.N., Haskell B.G., Digital pictures: representation and compression, New York, Plenum 1988. 3. Chan C-K., Cheng L.M., Hiding data In images by simple LSB substitution, Pattern Recognition 37, 2004. – S. 469–474 4. Zhang X., Wang S., Vulnerability of pixel – value differencing steganography to histogram asalysis and modification for enhanced security, Pattern Recognition Letters 25, 2004. – P. 331–339 5. http://www.ws.binghamton.edu/fridrich/Research/stochastic_modulation02.pdf