УДК 004.056.5

І.І. Борисенко

ЗАСТОСУВАННЯ ТЕОРІЇ ГРАФІВ В ЗАДАЧАХ СТВОРЕННЯ СТЕГАНОГРАФІЧНИХ ПОВІДОМЛЕНЬ

У статті розглядається новий стеганографічний алгоритм просторової області вбудовування в цифрове зображення, заснований на теорії графів. Основним принципом розробки є мінімізація впливів вбудованого повідомлення на контейнер. В основу алгоритму покладено обмін місцями елементів контейнера в більшій мірі, ніж їх модифікація. Алгоритм дозволяє зменшити викривлення контейнера, зберегти статистики першого порядку та забезпечити стійкість до найбільш відомих статистичних атак.

Ключові слова: *стеганографічний алгоритм, повідомлення, контейнер, викривлення контейнера.*

В статье рассматривается новый стеганографический алгоритм пространственной области встраивания в цифровое изображение, который базируется на теории графов. Основным принципом разработки является минимизация влияния встроенного сообщения на контейнер. В основу алгоритма положено обмен местами элементов контейнера в большей степени, чем их модификация. Алгоритм позволяет уменьшить искажения контейнера, сберечь статистики первого порядка, а также обеспечить устойчивость к наиболее известным статистическим атакам.

Ключевые слова: *стеганографический алгоритм, сообщение, контейнер, искажение контейнера.*

A new steganographic algorithm for the spatial-domain of digital images, based on the graph theory is suggested n the paper. The main design principle is to minimize an embedding impact by means of efficient coding algorithm. The basis for algorithm is the idea of exchanging rather than overwriting pixels. The algorithm allows reducing the distortions of the container, as well saving the first order statistic and providing the stability against the statistical attacks.

Keywords: *steganographic algorithm, message, container, embedding impact.*

Основною метою використання комп'ютерної стеганографії є приховування повідомлень в цифрових даних (ЦД), які, як правило, мають аналогову природу (мова, зображення, аудіо або відеозапис). Це ефективний засіб захисту інформації, який стає особливо актуальним у випадку, коли застосування криптографічних методів неможливо або обмежено. В якості повідомлення виступає будь-яка конфіденційна інформація (особисті та медичні дані, банківська та комерційна інформація і т.п.), яка повинна бути вбудована таким чином, щоб навіть сам факт її присутності у контейнері був таємним. ЦД, в які вбудовується повідомлення носять узагальнену назву – контейнер, результатом такого вбудовування є стеганоконтейнер або стеганографічне повідомлення, яке відкрито пересилається одержувачу каналами загального користування [1, 2].

З зростанням кількості і складності стеганографічних методів та алгоритмів приховування повідомлень зростає також і кількість методів стеганоаналізу, які переслідують мету виявлення стеганографічних вкладень.

Програмні інструменти приховування інформації такі як Steganos, Outguess, Jsteg, Jphs, S-Tools та інші прості в використанні і здатні створити стеганографічний канал з великою пропускною здатністю. Ці інструменти, як правило, використовують метод найменшого значущого біта (LSB) [2] та його модифікації, але стеганоконтейнери, які створені за допомогою LSB, успішно виявляються методами стеганоаналізу. Цей факт призвів до появи цілого ряду робіт, присвячених методам вбудовування в молодший біт без суттєвого порушення закону розподілення бітів. Наприклад, тривіальною модифікацією методу LSB-replacement є LSB-*matching*, який випадковим чином змінює піксельні значення на ± 1 так, що молодші біти пік селів відповідають бітам повідомлення, що вбудовується. Завдяки такій модифікації LSB-*matching* стеганоконтейнери набагато важче розпізнаються методами стеганоаналізу, тому це призвело до появи ряду робіт, в яких досліджується стійкість цього алгоритму [3–6].

Окрему групу методів, які схожі своєю спільною ідеєю, складають так звані методи мінімального стеганографічного збурення [7-9], застосування яких значно підвищує стійкість стеганографічних систем.

Вбудовування повідомлення відбувається за рахунок корегування елементів контейнера, що призводить до зміни його характеристик. Саме ці зміни в характеристиках використовуються методами стеганоаналізу для розпізнавання стеганоконтейнерів: статистичні характеристики (послідовна кореляція, ентропія, статистики першого порядку [10-12], характеристики спектру матриці контейнера (сингулярні та власні числа) [13] та ін. Зрозуміло, що чим менше збурень зазнає контейнер під час вбудовування повідомлення, тим важче стеганоаналітичним методам забезпечити низький рівень похибки при розпізнаванні. Отже, якщо забезпечити обмін елементів контейнера, а не їх модифікацію, то тим самим будуть збережені статистичні характеристики контейнера.

Метою роботи є підвищення ефективності передачі стеганографічних повідомлень шляхом розробки нового стеганографічного алгоритму, заснованого на теорії графів, стійкого до статистичних атак за рахунок збереження статистик першого порядку контейнера.

Для досягнення мети було поставлено такі завдання:

1) розробка графової моделі контейнера;

2) розробка нового стеганографічного алгоритму в рамках побудованої моделі;

3) оцінка стійкості розробленого стеганоалгоритму до статистичних атак;

4) оцінка збурень контейнера після вбудовування повідомлення.

В основі побудови моделі контейнера лежить побудова його графа, тому наведемо декілька понять з теорії графів [14, 15]. Граф G представляє собою структуру (V, E) де V – множина його вершин, а $E \subseteq V \times V$ – множина ребер. Неорієнтований граф – це граф, усі ребра якого не мають орієнтації, тобто $(x, y) \in E$ і $(y, x) \in E$ – це одне і те саме ребро. Дві вершини суміжні, якщо вони з'єднані ребром, тобто, якщо $(y, x) \in E$ – це ребро, то y та x суміжні. Два ребра називаються суміжними, якщо вони інциндентні одній і тій вершині, тобто такі ребра мають спільну вершину. Будь-який граф однозначно визначається матрицею

суміжності. Матриця суміжності А з елементами a_{ij} – це матриця виду: $a_{ij} = \begin{cases} 0, \text{ якщо } (x, y) \in E \\ 1, \text{ якщо } (x, y) \notin E \end{cases}$.

Паросполученням (або незалежною множиною ребер) графа G називають множину ребер у якій ніякі два ребра не суміжні. Паросполучення графа G називають максимальним, якщо воно не міститься в жодному парасполученні з більшою кількістю ребер, і найбільшим, якщо кількість ребер у ньому найбільша серед усіх парасполучень графа G.

Побудова графової моделі контейнера передбачає:

– розбивку множини елементів контейнера на групи, що не перетинаються;
– визначення способу кодування кожної групи;
– конструювання вузлів графа;
– визначення способу представлення графа.

Контейнер, який буде використовуватися для вбудовування інформації позначимо літерою С, а його елементи -s_i, тоді $C = \{s_1, ..., s_i, ..., s_n\}$, де n – кількість елементів контейнера, $s_i \in S$, де S – множина значень елементів контейнера.

Визначимо функцію $f : S \to \{0, ..., p-1\}$, яка кожному ставить у відповідність елемент з множини $\{0, ..., p-1\}$. Для вбудовування одного елемента повідомлення будемо використовувати не один елемент контейнера, а групу з c елементів так, як це запропоновано в [16, 17]. Такий підхід дає більшу свободу вибору одного елемента з для модифікації. Таким чином, після розбивки елементів контейнера на $k = \left\lfloor \frac{n}{c} \right\rfloor$ ($\lfloor \bullet \rfloor$ – ціла частина $\bullet$) неперетинаючихся груп одержимо структуру $C' = \{c_1, ..., c_i, ..., c_k\}$, де $c_i = (s_{i1}, ..., s_{ic})$. Будь-яке $c_i' = f(f(s_{i1}) + ... + f(s_{ic}))$ визначає деяке значення, яке вже присутнє в контейнері.

Повідомлення, позначимо його $M = \{m_1, ..., m_i, ..., m_k\}$ кодуються за тим же принципом, що й елементи контейнера, тобто $m_i \in \{0, ..., p-1\}$. Наприклад, якщо p=2, то значеннями m_i є біти.

Вузол графа представляє собою структуру $v_i(X_i, Y_i)$, де $Y_i = (y_{i1}, ..., y_{ic})$ – i-та група цільових значень, а $X_i = (x_{i1}, ..., x_{ic})$ – i-та група позицій елементів контейнера, які складають Y_i. Вузли створюються тільки у випадку, коли $m_i \neq c_i'$. Використання паросполучень побудованого графа дасть можливість зменшити кількість корегованих елементів контейнера і вбудовування інформації буде в більшій мірі відбуватися за рахунок їх обміну.

Як вже відмічалося, вузли графа створюються тільки у випадку, коли $m_i \neq c_i'$, але, не зважаючи на це, їх кількість є значною. Так, наприклад, при вбудовуванні повідомлення розміром 1КВ в середньому маємо три тисячі вузлів і до десяти тисяч ребер, а при збільшенні розміру вкладень до 4КВ кількість вузлів відповідно збільшується в чотири рази, а кількість ребер сягає півтора мільйони. Тому автори [16] відмовилися від використання списків суміжності, а запропонували дві структури, які також є громіздкими.

В цій роботі пропонується контейнер розбити на блоки і для кожного блоку будувати граф, розмір якого дозволить використовувати його матрицю суміжності, яка однозначно представляє граф.

Процес побудови графа представимо схемою (рис. 1), взявши конкретні значення параметрів графової моделі c=3 та p=4 [16].

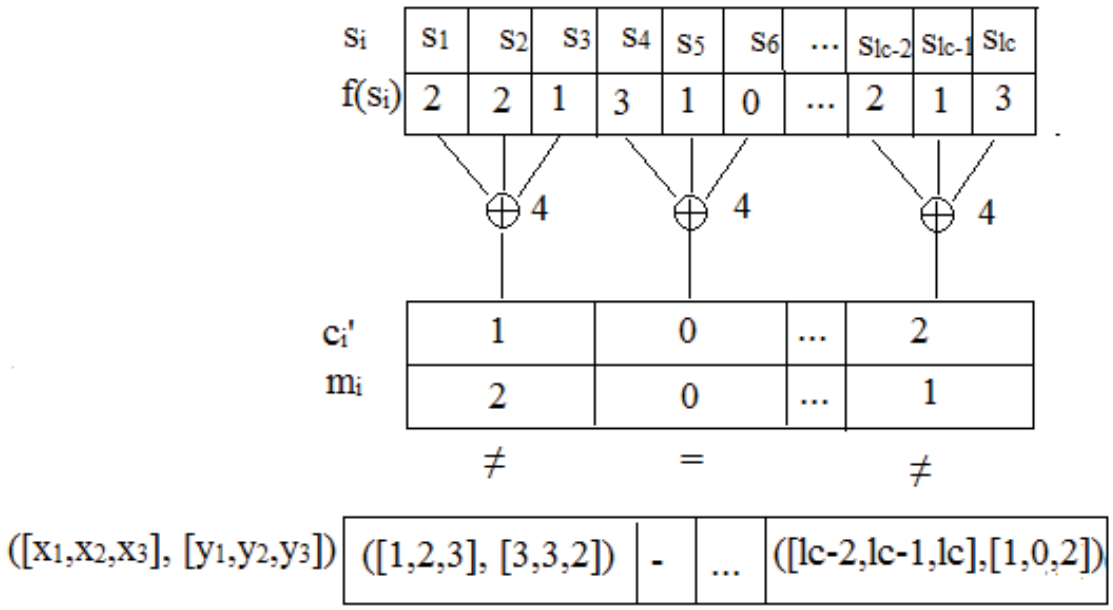

Рис. 1. Схема побудови графа

Перша стрічка містить елементи контейнера. Друга – значення, визначені функцією *f*. Три (в загальному випадку *c*) елемента $f(s_i)$ модифікуються завдяки застосуванню операції додавання по модулю 4 (в загальному випадку по модулю *p*) щоб одержати значення c_i', які порівнюються з елементами повідомлення m_i. Якщо значення не співпали, як в першому та останньому випадку, то створюється вузол, як показано в останній стрічці. Цільові значення вузла [y_1, y_2, y_3] обчислюються додаванням різниці $m_i - c_i$ до кожного значення $f(s_i)$. Заміна одного з $f(s_i)$ на його цільове значення приведе у відповідність c_i' та m_i.

Розглянемо як створюються ребра. Вище було зазначено, що алгоритм вбудовування переслідує мету в більшій мірі обмінювати елементи контейнера, ніж модифікувати. Розглянемо перший і останній вузли схеми, представленої на рисунку 1. Цільове значення елемента s_1 дорівнює трьом при значенні $f(s_i)$=2, а елемента s_{lc} дорівнює двом при $f(s_{lc})$=3. Якщо різниця їх значень $d = |s_1 - s_{lc}|$ дозволяє обміняти ці елементи місцями, тобто обмін не призведе до видимого (якщо контейнер – це зображення) спотворення контейнера, то створюється ребро. Легко помітити, що ребер між двома вузлами може бути декілька, наприклад, також можна створити ребро (s_2, $s_{lc}-2$) за умови виконання вимоги до різниці їх значень. Це дає більше ступенів свободи для вибору з цих ребер одного, виходячи, наприклад, з мінімальності значення *d*.

Стеганографічний алгоритм, назвемо його GRAPH_*matching* представимо наступними кроками.

Крок 1. Розбити матрицю контейнера – зображення на блоки заданого розміру.

Крок 2. Для кожного блоку побудувати граф G_i, виконуючи дії представлені схемою, зображеною на рисунку 1.

Крок 3. У графі G_i визначити найбільше паросполучення.

Крок 4. Для вузлів, які належать найбільшому паросполученню виконати обмін елементів контейнера.

Крок 5. Для вузлів, які не належать найбільшому паросполученню виконати модифікацію елементів контейнера.

Щоб декодувати повідомлення, вбудоване алгоритмом GRAPH_ *matching* треба розбити матрицю стеганоконтейнера на блоки того самого розміру, що і при вбудовуванні повідомлення. Використовуючи ті самі значення для параметрів c і p, які використовувалися при вбудовуванні, обчислити $f(s_i)$ та c_i'. Значення c_i є елементами повідомлення.

Щоб оцінити стійкість розробленого стеганоалгоритму до статистичних атак застосовувався метод оцінки числа переходів значень молодших бітів в сусідніх елементах [10] (рис.2) та гістограм ний метод (рис. 3).

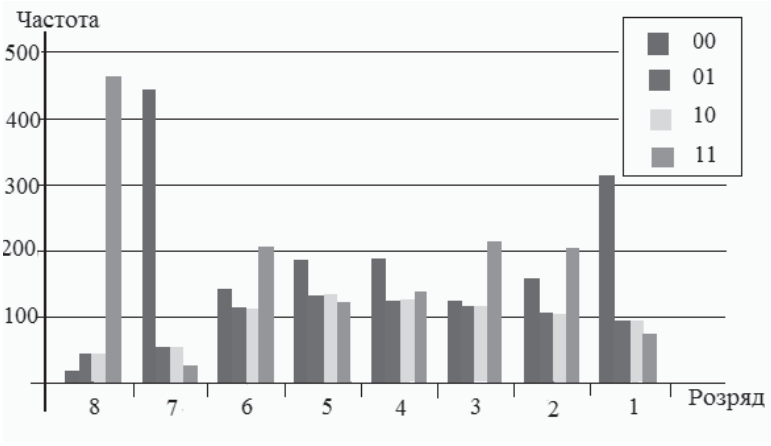

Рисунок 2. Гістограма частот переходів бітових значень

Число переходів виду 0 в 0, 0 в 1, 1 в 0, 1 в 1 в потоці бітових значень стеганоконтейнера не повинно бути випадковим (у цьому разі всі частоти в розряді будуть однаковими), оскільки це не властиво реальному контейнеру. З огляду на гістограму частот (рис. 2), одержано стеганоконтейнер, який нечутливий до означеного методу.

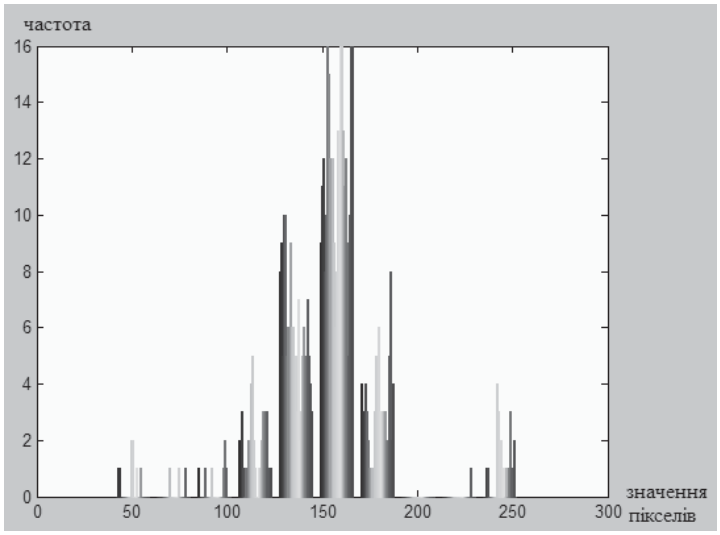

Рис. 3. Гістограма стеганоконтейнера, метод GRAPH_ *matching*

Гістограма контейнера не приведена в роботі, оскільки її вигляд співпадає з гістограмою рис. 3. Саме цей результат і підтверджує поставлену мету, тобто збережені статистики першого порядку контейнера.

За рахунок того, що основна кількість елементів контейнера обмінюється місцями, а не корегується зберігаються його статистичні характеристики, але з точки зору збурень, яких зазнає контейнер представлений алгоритм є вразливим до методів стеганоаналізу, які ці збурення відслідковують. Алгоритм GRAPH_*matching* можна зробити більш ефективним, якщо модифікувати крок 4. Вбудовування повідомлення в контейнер надалі будемо називати стеганоперетворенням (СП). СП контейнера з матрицею можна представити як збурення вихідної матриці С ($\overline{C} = C + \Delta C$, де $\overline{C}$ − матриця стеганоконтейнера (СК)), або у вигляді сукупності збурень множини сингулярних чисел (СНЧ) та сингулярних векторів матриці контейнера, які однозначно її визначають [18]. Норма матриці збурень $\|\Delta C\|_2$ не залежить від того, які саме СНЧ були збурені, а залежить лише від абсолютних величин цих збурень, тому в подальшому будемо використовувати саме СНЧ.

В залежності від того, які елементи матриці контейнера будуть модифікуватися під час СП, рівень збурень матриці контейнера буде неоднаковим. Тому пропонується зробити попередній аналіз елементів контейнера з метою визначення кількісної оцінки (позначимо її μ_{ij}) вкладу кожного конкретного елемента C_{ij} контейнера у $\|\Delta C\|_2$, на випадок, якщо елемент зазнає модифікації при СП. Збурення елементів контейнера будемо моделювати найменш можливим значенням +1 [19].

Основні кроки обчислення μ_{ij} та її використання:

1) збурити елемент C_{ij}, в результаті одержуємо матрицю $\overline{C}_{\sim ij}$, в якій усі елементи співпадають з елементами матриці C, окрім одного, значення якого змінилося на одиницю;

2) для матриць С та $\overline{C}_{\sim ij}$ побудувати нормальний сингулярний розклад [18]: $C = USV^T$, $\overline{C}_{\sim ij} = \overline{U}_{\sim ij}\overline{S}_{\sim ij}\overline{V}_{\sim ij}$;

3) знайти збурення матриці СНЧ: $\Delta S = S - \overline{S}_{\sim ij}$;

4) оцінити значення: $\mu_{ij} = \max_i |\Delta S_{ii}|$, де ΔS_{ii} − діагональні елементи матриці ΔS;

Таким чином результатом попередньої обробки матриці контейнера є матриця M значень μ_{ij}. Як уже зазначалося, що ребер між двома вузлами графа може бути декілька, тому таку ситуацію використовуємо слідуючим чином.

Крок 4 (модифікований) алгоритму GRAPH_*matching*

Якщо вузлу інцидентно декілька ребер, то для обміну елементів s_{ij} та s_{kl}, вибрати ту пару, якій відповідає найменша сума $\mu_{ij} + \mu_{kl}$.

Для прикладу повернемося до рис. 1. Щоб c_i' було рівним значенню 2 для першого вузла, що відповідає значенню відповідного елемента повідомлення, можна поміняти місцями s_1 з s_k або s_2 з s_k або s3 з s_{k-2}, перевагу віддаємо тій парі, для яких сума відповідних їм μ найменша.

Проведемо оцінку збурень контейнера, які викликані вбудовуванням повідомлення алгоритмом GRAPH_*matching* та його модифікованою версією GRAPH_*matching_1*.

Таблиця 1

Порівняльна характеристика збурень контейнера

Об'єм ДІ	GRAPH_*matching_1*	GRAPH_*matching*
10 бітів	1,4142	2,1358
1/5 контейнера	34,6888	45,3085
1/2 контейнера	49,3305	65,8341

Одержані дані свідчать про наявність ефекту від застосування попередньої обробки матриці контейнера.

Висновки

В роботі побудовано стеганографічний алгоритм GRAPH_*matching*, заснований на теорії графів. В основу алгоритму покладено принцип обміну елементів контейнера при вбудовуванні повідомлення в більшій мірі, ніж їх корегування. За рахунок того, що основна кількість елементів контейнера обмінюється місцями, а не корегується зберігаються його статистичні характеристики, що робить його стійким до статистичних атак. З точки зору збурень, яких зазнає матриця контейнера при такому способі вбудовування алгоритм GRAPH_*matching* нічим не відрізняється від інших алгоритмів, які базуються на корегуванні елементів контейнера. Цей недолік частково долається шляхом приписування, кожному елементу контейнера деякого коефіцієнта, який відображує внесок кожного елемента в загальний рівень збурення контейнера при СП, а потім використовується алгоритмом GRAPH_*matching*_1.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Хорошко В.О.* Основи комп'ютерної стеганографії : навч. посіб. для студентів і аспірантів / Хорошко В.О., Азаров О.Д., Шелест М.Є., Яремчук Ю.Є. – Вінниця : ВДТУ, 2003. – 143 с.

2. *Грибунин В.Г.* Цифровая стеганография / В.Г. Грибунин, И.Н. Оков, И.В. Туринцев. – М. : Солон-Пресс, 2002. – 272 с.

3. *Ker A.D.* Stegaanalysis of LSB matching in grayscale images / Ker A.D. // IEEE Signal Processing Letters. – 2005. – Vol. 12. – № 6. – P. 441–444.

4. *Liu Q.Z.* Image complexity and feature mining for stegaanalysis of least significant bit matching steganography / Q.Z. Liu, A.N. Sung, et al. // Information Sciences. – 2008. – Vol. 178. – № 1. – P. 21–36.

5. *Zhihua Xia.* A Learning-Based Stegaanalytic Method against LSB Matching Steganography / Xia Zhihua, Lincong Yang, et al. // Radioengineering. – 2011. – Vol. 20. – № 1. – P. 102–109.

6. *Nataradjan V.* Blind Image Stegaanalysis Based on Contourlet Transform / V. Nataradjan, R. Anitha // International Journal on Cryptography & Information Security. – 2012. – Vol. 2. – Iss. 3. – № 1. – P. 77–87.

7. *Filler T.* Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes / T. Filler, J. Judas, J. Fridrich // Forensics and Security. – 2011. – Vol. 6(1). – P. 920–935.

8. *Kodovskə J.* On Dangers of Overtraining Steganography to an Incomplete Cover Model / J. Kodovskə, J. Fridrich, V. Holub // Proc. ACM Multimedia & Security Workshop (Niagara Falls, New York, September 29–30, 2011). – P. 69–76.

9. *Filler T.* Gibbs construction in Steganography / T. Filler, J. Fridrich // Forensics and Security. – 2010. – № 5(4). – P. 705–720.

10. *Барсуков В.С.* Оценка уровня скрытности мультимедийных стеганографических каналов хранения и передачи информации / В.С. Барсуков, А.П. Романцов // Специальная Техника. – 2000. – № 1.

11. *Дрюченко М.А.* Алгоритмы выявления стеганографического скрытия информации в jpeg-файлах / М.А. Дрюченко // Системный анализ и информационные технологии. – 2007. – № 1. – С. 21–30.

12. *Иванов М.А.* Теория, применение и оценка качества генераторов псевдослучайных последовательностей / М.А. Иванов, И.В. Чугунков. – М. : КУДИЦ-ОБРАЗ, 2003.

13. *Бобок И.И.* Детектирование наличия возмущений матрицы цифрового изображения как составная часть стеганоанализа / И.И. Бобок // Вісн. Східноукр. нац. ун-ту ім. В. Даля. – 2011. – № 7(161). – С. 32–41.

14. *Харари Ф.* Теория графов / Ф. Харари. – М. : Мир, 1993. – С. 203.

15. *Нікольський Ю.В.* Дискретна математика / Ю.В. Нікольський, В.В. Пасічник, Ю.М. Щербина. – К. : Видавнича група BHV, 2007. – С. 354.

16. *Hetzl S.* A *Graph*-Theoretic Approach to Steganography in Proc. Communications and Multimedia Security / *S. Hetzl*, P. Mutzel. – 2005. – P. 119–128.

17. *Ross J. Anderson.* Stretching the Limits of Steganography / Ross J. Anderson, editor // Information Hiding, First International Workshop. – 1996. – Vol. 1174. – P. 39–48.

18. *Кобозева А.А.* Загальний підхід до оцінки властивостей стеганографічного алгоритму, заснованій на теорії збурень / А.А. Кобозева // Информационные технологии и компьютерная инженерия. – 2008. – № 1 (11). – С. 164–171.

19. *Борисенко І.І.* Мінімізація збурень контейнера при його стеганографічному перетворенні / І.І. Бориченко // Матеріали 3-ої МНПК ІУСТ. – 2014. – С. 199–201.