

УДК 004.932 : 621.391.7

СПОСІБ СТЕГАНОГРАФІЇ ЗОБРАЖЕНЬ З ФРАГМЕНТАЦІЄЮ СТЕГОДАНИХ ТА РОЗДІЛЕННЯМ ЗАКРИТОГО КЛЮЧА

Євгенія Сулема, Семен Широчин

НТУУ «КПІ», Факультет прикладної математики

Анотація: Запропонований спосіб відноситься до стеганографії простору зображень та ґрунтується на розміщенні стегоданих у найменш значущих бітах з використанням спеціального алгоритму фрагментації. Спосіб передбачає утворення закритого ключа, що може бути розділений на частини для збереження та передачі їх окремо від контейнера.

Summary: The proposed method deals with image domain steganography and is based on less significant bit stegodata allocation with special fragmentation algorithm. The method implies a separable private key to keep it separately from the container.

Ключові слова: Стеганографія, фрагментація, закритий ключ, стеганографічний контейнер.

Вступ

Наразі існує багато методів захисту інформації, в тому числі й таких, що ґрунтуються на стеганографії. Стеганографічний спосіб захисту передбачає, що дані з обмеженим доступом (*стегодані*) вбудовуються у файл (*контейнер*), який знаходиться у відкритому доступі [1]. Цифрова стеганографія передбачає використання як контейнера графічних, аудіо- та відео-файлів [2].

Найбільш поширеним і водночас перспективним видом стеганографії є стеганографія зображень. Це пов'язано з такими особливостями зображень, як надлишковість графічних даних та достатній для приховування стегоданих об'єм файлу. Стеганографія зображень реалізується за рахунок розміщення стегоданих в найменш значущих бітах контейнера (Less Significant Bit, LSB) [1]. Для зберігання стегоданих може використовуватись різна кількість молодших бітів відповідно до обсягу даних, що приховуються, та залежно від графічного вмісту контейнера [3]. При цьому має гарантовано забезпечуватись відсутність візуальних спотворень зображення-контейнера. На сьогоднішній день існує багато форматів графічних файлів, які підтримують ущільнення без втрат і таким чином дозволяють гарантувати відсутність спотворень стегоданих [2]. Найкращими контейнерами для LSB-стеганографії є неархівовані і неформатовані зображення з великою кількістю кольорів [4].

Зображення мають значний потенціал для їх використання в ролі стеганографічного контейнера. Оскільки методи стегоаналізу постійно вдосконалюються, то задача створення нових способів стеганографічного захисту інформації є та буде залишатись актуальною задачею.

I Мета дослідження та постановка задачі

Загальною метою дослідження є створення способів захисту графічних даних при їх передаванні відкритими каналами зв'язку. В рамках даної статті вирішується задача прихованої передачі графічних стегоданих, які вбудовуються в графічний файл-контейнер, із забезпеченням виконання наступних умов:

- використання як контейнера для стегоданих файлу одного із поширених графічних форматів;
- максимально непередбачуване розміщення блоків стегоданих для ускладнення стегоаналізу.

Крім того, спосіб стеганографічного захисту графічних даних має бути незалежним від конкретного графічного формату [2].

II Схема стеганографічного перетворення

Для вирішення поставленої задачі пропонується спосіб, що відноситься до стеганографії вмісту зображень (Image Domain Steganography). Як контейнер для стегоданих будуть використовуватись графічні дані цього файлу. Метадані можуть бути застосовані для збереження частини закритого ключа. При цьому формат графічного файлу змінюватись не буде. Це дозволить виконати вимогу щодо незалежності способу стеганографічного захисту графічних даних, що розробляється, від конкретного формату даних.

Використання стегоданих, незахищених додатково, може призвести до несанкціонованого доступу до секретних даних в разі виявлення їх наявності в контейнері. Тому на практиці стегодані звичайно захищають криптографічним ключем. Як альтернатива криптографічному захисту стегоданих в даній статті пропонується виконувати фрагментацію стегоданих. У разі застосування фрагментації секретом стає не стільки наявність прихованих даних, скільки схема їх розташування у контейнері й алгоритм, що дозволить отримати їх правильну послідовність і відновити зображення [1].

Пропонується стеганографічна схема (рис. 1), в якій даними, необхідними для відновлення вихідного зображення, є наступні:

- множина вказівників на адресу початку розташування кожного блоку стегоданих (ключ 1);
- множина довжин блоків стегоданих (ключ 2).

Ці дані являють собою закритий стеганографічний ключ, який може бути розташований в блоці метаданих результуючого контейнера в зашифрованому вигляді або зберігатися окремо від файлу.

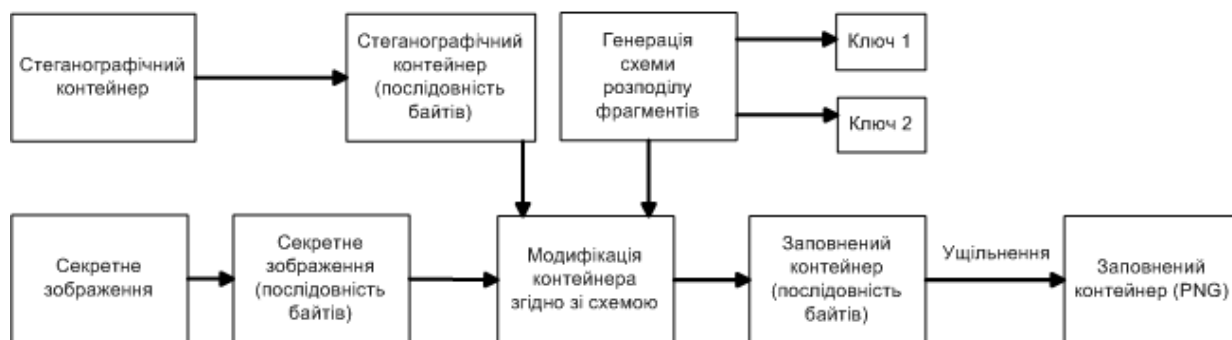


Рисунок 1 – Схема стеганографічного перетворення даних з фрагментацією

Оскільки передача заповненого контейнера буде відбуватись через мережу Internet, то за контейнер потрібно обирати файл такого формату, який забезпечить ущільнення даних без втрат. При цьому саме ущільнення покладається на кодек обраного формату. В рамках даної статті передбачається використання формату PNG, що застосовує комбінацію алгоритмів ущільнення Зев-Лемпеля та Хафмана, а також дельта-фільтрацію. Використання дельта-фільтрації, що замість значення байта записує різницю його значення з попереднім, створює велику кількість бітів з однаковими значеннями, що робить вихідний файл більш придатним для ущільнення:

$$\begin{cases} B[i] = A[i], i = 0 \\ B[i] = A[i] - A[i - 1], i > 0, \end{cases} \quad (1)$$

де B – масив значень байтів результуючого файлу,
 A – масив значень байтів початкового файлу.

III Алгоритм розміщення блоків стегоданих

До алгоритму розміщення блоків стегоданих висуваються наступні вимоги [5].

1. Дані повинні бути фрагментовані, щоб їх неможливо було отримати простим зчитуванням.
2. Послідовність фрагментів має бути максимально непередбачуваною, щоб не можна було легко виявити закономірність їх розташування (в ідеальному випадку – забезпечення відсутності логіки розміщення даних).
3. Розміри фрагментів мають бути неоднаковими і варіюватись у деякому діапазоні. В такому випадку стегоаналіз не може бути обмежений пошуком можливих розташувань блоків однакового розміру.
4. Щільність фрагментів має бути неоднорідною, щоб унеможливити обчислення закономірностей відстаней між блоками.
5. Розмір контейнера має впливати на розподіл діапазону розмірів фрагментів; чим менше місця в контейнері залишається для звичайних даних, тим дрібнішими мають бути блоки стегоданих.

Для забезпечення виконання цих умов пропонується алгоритм, що ґрунтується на застосуванні випадкових чисел та алокації (контролі за накладанням блоків даних).

Алгоритм фрагментації блоків має наступний вигляд. Для кожного блоку генерується випадкове значення адреси:

$$B_i = \gamma(C), \quad (2)$$

де i – номер блоку даних;

C – максимальне значення, яке може прийняти адреса, що дорівнює об'єму контейнера.

При цьому має виконуватись умова незайнятості даної адреси:

$$\begin{cases} B_i > B_k + S_k, \\ B_i < B_j, \end{cases} \quad (3)$$

де B_i – початкова адреса i -го блоку даних;

B_j – початкова адреса найближчого наступного блоку;

B_k – початкова адреса найближчого попереднього блоку;

S_k – довжина найближчого попереднього блоку;

У випадку невідповідності умові (3), відбувається повторне обчислення значення B_i за формулою (2).

Після обчислення адреси початку блоку даних обчислюється випадкове значення розміру даного блоку:

$$S_i = \gamma(D), \quad (4)$$

де i – номер блоку даних;

D – максимальне значення довжини блоку, що дорівнює об'єму нефрагментованих секретних даних.

При цьому має виконуватись вимога можливості розміщення блоку такої довжини за цією адресою:

$$B_i + S_i < B_j, \quad (5)$$

де B_i – початкова адреса i -го блоку даних;

S_i – довжина i -го блоку даних;

B_j – початкова адреса найближчого наступного блоку даних.

Об'єднавши правила (4) і (5), можна сформулювати загальне правило генерації випадкової величини адреси i -го блоку:

$$S_i = \gamma(\min(D, B_j - B_i)). \quad (6)$$

Таким чином, випадкове число довжини блоку водночас обмежене об'ємом нефрагментованих даних і початковою адресою найближчого наступного блоку даних.

Наприкінці кожної ітерації об'єм нефрагментованих даних зменшується на величину довжини блоку. Створення нових блоків припиняється, коли не залишається нефрагментованих даних. Схематично алгоритм фрагментації блоків зображено на рис. 2.

Отриманий ключ складається з двох чисельних векторів: вектора $\bar{a}[n]$ початкових адрес та вектору $\bar{l}[n]$ довжин фрагментів, де n – довжина векторів, що відповідає кількості фрагментів. Для збереження кожного фрагменту потрібна кількість пам'яті, яка зможе вмістити число, що буде відповідати початковій адресі фрагменту.

Якщо враховувати, що чисельний тип *unsigned long int* дозволяє представляти числа в діапазоні від 0 до 4 294 967 295, то використовуючи цей тип, можна записати будь-яку адресу в межах файла обсягом до 4-х Гб. Тип *unsigned long* займає 4 байти, отже об'єм отриманого ключа буде залежати від кількості фрагментів і визначатись за формулою:

$$S(\bar{a}) = S(\bar{l}) = 4n, \text{ байт}, \quad (7)$$

де n – кількість утворених фрагментів.

Пропонуються наступні варіанти щодо місця збереження ключа:

- в блоці метаданих контейнера;
- як стегодані іншого зображення;
- як стегодані інших зображень (окремі вектори).

Для посилення захисту даних, що приховуються, рекомендується передавати вектори ключа (ключ 1 та ключ 2 на рис. 1) окремими повідомленнями.

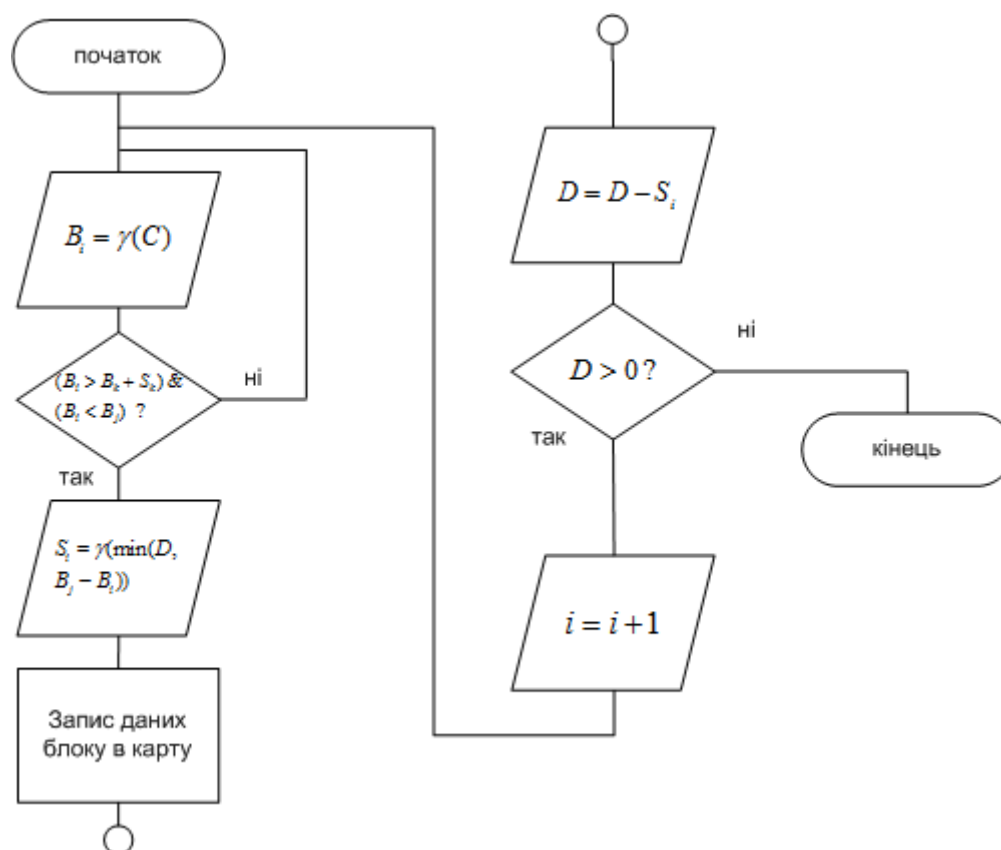


Рисунок 2 – Алгоритм фрагментації блоків

IV Відновлення стегоданих

Для отримання початкового секретного повідомлення потрібен контейнер з вбудованими стегоданими, а також розділений стеганографічний ключ, що складається з вектора \bar{a} початкових адрес та вектора \bar{l} довжин фрагментів. Послідовність значень у векторі відповідає послідовності фрагментів у секретному повідомленні. Процес відновлення складається з наступних дій.

1. Зчитування векторів \bar{a} та \bar{l} та перевірка ідентичності їх довжин.
2. Перехід до наступного елемента вектора \bar{a} та зчитування фрагменту довжиною \bar{l} .
3. Конкатенація результуючого вектора \bar{x} і зчитаного фрагмента.
4. Якщо вектори \bar{a} та \bar{l} ще не закінчились, перехід до п. 2.
5. Збереження вектора \bar{x} у вихідний файл.

Алгоритм відновлення вихідного секретного повідомлення описаний формулою (8):

$$X[i] = C[a[i]] \cdot C[a[i] + 1] \cdot \dots \cdot C[a[i] + l[i]], \quad (8)$$

де – номер блоку,

$l[i]$ – i -й елемент вектора l ,

$a[i]$ – i -й елемент вектора a ,

C – масив байтів контейнера,

“.” – конкатенація масиву байтів блоку стегоданих.

V Аналіз швидкодії стеганографічних перетворень

Очевидно, що застосування фрагментації уповільнює процес перетворення стегоданих. Для оцінки швидкодії алгоритму, що реалізує запропонований спосіб стеганографії зображень з фрагментацією стегоданих, він був реалізований програмним шляхом та протестований на різних комбінаціях зображень-оригіналів та зображень-контейнерів. Аналогічно був реалізований та протестований алгоритм без фрагментації стегоданих. Швидкодія визначалась для обох напрямків роботи алгоритму (табл. 1 та табл. 2):

- 1) формування заповненого контейнера даними вхідного зображення (запис стегоданих);
- 2) відтворення зображення (зчитування стегоданих).

При записі стегоданих загальний час виконання алгоритма (табл. 1) включає в себе час, необхідний на генерування карти фрагментів, модифікацію бітів контейнера та експорт ключів.

При зчитуванні стегоданих загальний час виконання алгоритму (табл. 2) включає в себе час, необхідний на імпорт ключів, зчитування блоків та відновлення вихідного зображення.

Таблиця 1 – Час запису стегоданих (мс) без фрагментації (ліворуч) та з фрагментацією (праворуч)

Зображення Контейнер	Зображення 1 (320 x 213)	Зображення 2 (800 x 524)	Зображення 3 (1024 x 685)
Контейнер 1 (800 x 1024)	84 / 93	122 / 115	116 / 130
Контейнер 2 (1324 x 2048)	102 / 126	200 / 235	211 / 237

Таблиця 2 – Час зчитування стегоданих (мс) без фрагментації (ліворуч) та з фрагментацією (праворуч)

Оригінал Контейнер	Зображення 1 (320 x 213)	Зображення 2 (800 x 524)	Зображення 3 (1024 x 685)
Контейнер 1 (800 x 1024)	54 / 56	70 / 81	72 / 81
Контейнер 2 (1324 x 2048)	70 / 90	145 / 160	150 / 171

В середньому час запису стегоданих при використанні фрагментації збільшується на 11,5%, час зчитування – на 11,4%. Але абсолютне значення збільшення часу обробки стегоданих є несуттєвим (до 35 мс). Слід зазначити, що в табл. 1 та табл. 2 наведено значення часу, отримані лише для частини використаних зображень. Всього було проаналізовано 60 різних пар зображень «оригінал»-«контейнер».

Висновок

Запропонований стеганографічний спосіб відноситься до стеганографії вмісту зображень і використовує LSB для зберігання стегоданих. Особливістю запропонованого способу є використання альтернативного засобу захисту стегоданих, що полягає в фрагментації стегоданих в контейнері. Для відновлення секретного повідомлення використовується секретний ключ, що може бути збережений двома окремими частинами. Пропонується використання стандартного формату зображень PNG, що передбачає ущільнення графічних даних без втрат. Середній час роботи алгоритму з фрагментацією стегоданих перевищує час роботи алгоритму без фрагментації на 11,5%, що можна вважати несуттєвим збільшенням, враховуючи абсолютне значення затримки, що вноситься процедурою фрагментації. Отже, фрагментація стегоданих може бути рекомендована до практичного застосування в стеганографічних схемах.

Література: 1. Zaidan, B. Stego-Image Vs Stego-Analysis System / B. Zaidan, A. Zaidan, A. Taqa, F. Othman // International Journal of Computer and Electrical Engineering. – No. 5, December, 2009. – Vol. 1. 2. Morkel, T. An overview of image steganography / T. Morkel, J. H. P. Eloff, M. S. Olivier. – University of Pretoria, 2002. 3. Kharrazi, Mehdi. Image Steganography: Concepts and Practice / Mehdi Kharrazi, Husrev T. Sencar, Nasir Memon. – Polytechnic University of Brooklyn, USA. 4. Корольов, В. Ю. RS-Стегоаналіз. Принципи роботи, недоліки та концепція методу його обходу / В. Ю. Корольов, В. В. Поліновський, В. А. Герасименко // Вісник Вінницького політехнічного інституту, 2010. – № 6. – С. 71. 5. Сулема, Є. С. Критерії пошуку оптимального розташування блоків стеганографічних даних в контейнері / Є. С. Сулема, С. С. Широкин // Збірник тез доповідей міжнародної конференції «Системний аналіз та інформаційні технології», 2011. – С. 516.