УДК 004.932 : 621.391.7

К.т.н., доцент Сулема Є.С., студент Широчин С.С.

**Національний технічний університет України
«Київський політехнічний інститут»**

АЛГОРИТМ ФРАГМЕНТАЦІЇ СТЕГОДАНИХ У СТЕГАНОГРАФІЇ ЗОБРАЖЕНЬ

Abstract

Yevgeniya S. Sulema, assoc. prof., PhD; Semen Shyrochyn, student
Algorithm of stegodata fragmentation in image steganography

This paper is devoted to the algorithm of data fragmentation. The algorithm is developed to be used for an alternative stegodata protection in image steganography. The algorithm is randomized to make fragments order and size completely unpredictable. All the information, required to recover the order of fragments and their position inside of the container image, is hidden in the private key. The proposed method allows hidden data to be stored in public sites.

Вступ

Задача захисту інформації завжди була актуальною, але особливої важливості вона набула з розвитком комп'ютерних систем та мереж. Одним з методів захисту даних користувача при їх передачі по відкритих каналах зв'язку є стеганографія. Стеганографічний спосіб захисту передбачає приховування самого факту наявності секретної інформації у даних, що знаходяться у відкритому доступі. Одним з поширених способів стеганографічного захисту є стеганографія зображень, яка передбачає застосування несекретного графічного файлу як контейнера для приховування стегоданих, наприклад, графічного характеру (зображення). Оскільки в разі розкриття факту наявності стегоданих у контейнері, вони стають незахищеними від несанкціонованого доступу, то потрібно застосовувати додаткові способи захисту. Частіше за все застосовують криптографічні способи захисту даних. Але криптографічні алгоритми мають достатньо високу обчислювальну складність, тому задача пошуку альтернативних підходів до захисту стегоданих є актуальною. Дана стаття присвячена алгоритму фрагментації стегоданих, яка виконується з метою їх додаткового захисту.

Мета дослідження та постановка задачі

Метою дослідження є розроблення комплексного підходу до захисту зображень від несанкціонованого доступу при їх передаванні через мережу Інтернет та збереженні на серверах загального доступу.

В рамках даної статті розглядається вирішення задачі додаткового захисту стегоданих за рахунок їх фрагментації. Вимоги до алгоритму фрагментації, виконання яких дозволить ускладнити або навіть унеможливити стегоаналіз:

1. Послідовність фрагментів має бути максимально непередбачуваною, щоб не можна було виявити закономірність їх розташування.
2. Розміри фрагментів мають бути неоднаковими і варіюватись у деякому діапазоні, щоб унеможливити пошук фрагментів за фіксованим розміром.
3. Щільність фрагментів має бути неоднорідною, щоб унеможливити виявлення закономірностей відстаней між блоками.
4. Розмір контейнера має впливати на розподіл діапазону розмірів фрагментів: чим менше місця в контейнері залишається для звичайних даних, тим дрібнішими мають бути блоки стегоданих.

Додатковою вимогою є висока швидкодія алгоритму фрагментації. Загальний час стеганографічного перетворення при застосуванні фрагментації має бути порівнянним з часом стандартного стеганографічного перетворення, яке ґрунтується на використанні найменших значущих бітів (*Less Significant Bits, LSB*) контейнера для збереження стегоданих.

Таким чином, задача, вирішення якої розглядається в даній статті, зводиться до розробки алгоритму фрагментації стегоданих графічного характеру при їх збереженні у найменших значущих бітах контейнера (*LSB*-стеганографія зображень), який забезпечує ускладнення стегоаналізу та характеризується достатньо високою швидкодією.

Опис алгоритму

Для виконання умов, що висуваються до алгоритму фрагментації, пропонується застосувати генератор псевдовипадкових чисел. При цьому алгоритм, що використовує випадкові числа, має виконувати функцію алокатора, тобто контролювати неможливість перекриття блоків даних та гарантувати розміщення всіх блоків. Алгоритм фрагментації блоків стегоданих, що пропонується, складається з наступних дій.

1. Для кожного блока генерується випадкове значення адреси

$$B_i = \gamma(C), \qquad (1)$$

де i – номер блока даних;
C – максимальне значення адреси, що дорівнює об'єму контейнера.
1. Перевіряється умова незайнятості даної адреси:

$$\begin{cases} B_i > B_k + S_k \\ B_i < B_j \end{cases}, \qquad (2)$$

2. де B_i – початкова адреса i-го блока даних;

B_j – початкова адреса найближчого наступного блока;

B_k – початкова адреса найближчого попереднього блока;

S_k – довжина найближчого попереднього блока.

Якщо умова (2) не виконується, відбувається перехід на п. 1 алгоритму, тобто повторне обчислення значення B_i за формулою (1).
3. Обчислюється випадкове значення розміру даного блока:

$$S_i = \gamma(D), \qquad (3)$$

де *i* – номер блока даних;
D – максимальне значення довжини блока, що дорівнює об'єму нефрагментованих секретних даних.
4. Перевіряється умова можливості розміщення блока такої довжини за цією адресою:

$$B_i + S_i < B_j, \qquad (4)$$

де B_i – початкова адреса i-го блока даних;

S_i – довжина i-го блока даних;

B_j – початкова адреса найближчого наступного блока даних.

Об'єднавши правила (3) і (4), можна сформулювати загальне правило генерації випадкової величини адреси i-го блока:

$$S_i = \gamma(\min(D, B_j - B_i)) \qquad (5)$$

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

Наприкінці кожної ітерації об'єм нефрагментованих даних зменшується на величину довжини блока. Створення нових блоків припиняється, коли не залишається нефрагментованих даних. Таким чином, можна гарантувати як випадковий розподіл адрес і довжин, так і випадкову кількість блоків, на які будуть розділені стегодані. Обсяг результуючого контейнера буде дорівнювати об'єму початкового контейнера, що забезпечить ще одну важливу вимогу – вимогу "непідозрілості" файлу [1].

В результаті роботи алгоритму формуються наступні дані:
– заповнений контейнер;
– послідовність адрес блоків;

– послідовність довжин блоків.

Заповнений контейнер являє собою файл одного з графічних форматів (наприклад, *.bmp*, *.png* тощо) без ущільнення або з ущільненням без втрат. Послідовність адрес та послідовність довжин являють собою закритий ключ, який може бути переданий одним чи кількома окремими повідомленнями або бути вбудованим у метадані графічного файлу контейнера.

Аналіз запропонованого алгоритму

Для аналізу роботи алгоритму він був реалізований у вигляді програмного модуля та протестований на 60-ти комбінаціях контейнерів та зображень різних розмірів та графічного вмісту. Експериментальні дані свідчать про незначне збільшення часу, необхідного на стеганографічне перетворення. В середньому алгоритм з фрагментацією потребує на 11% часу більше, що з урахуванням факту забезпечення додаткового захисту стогоданих може вважатись прийнятним (рис. 1).

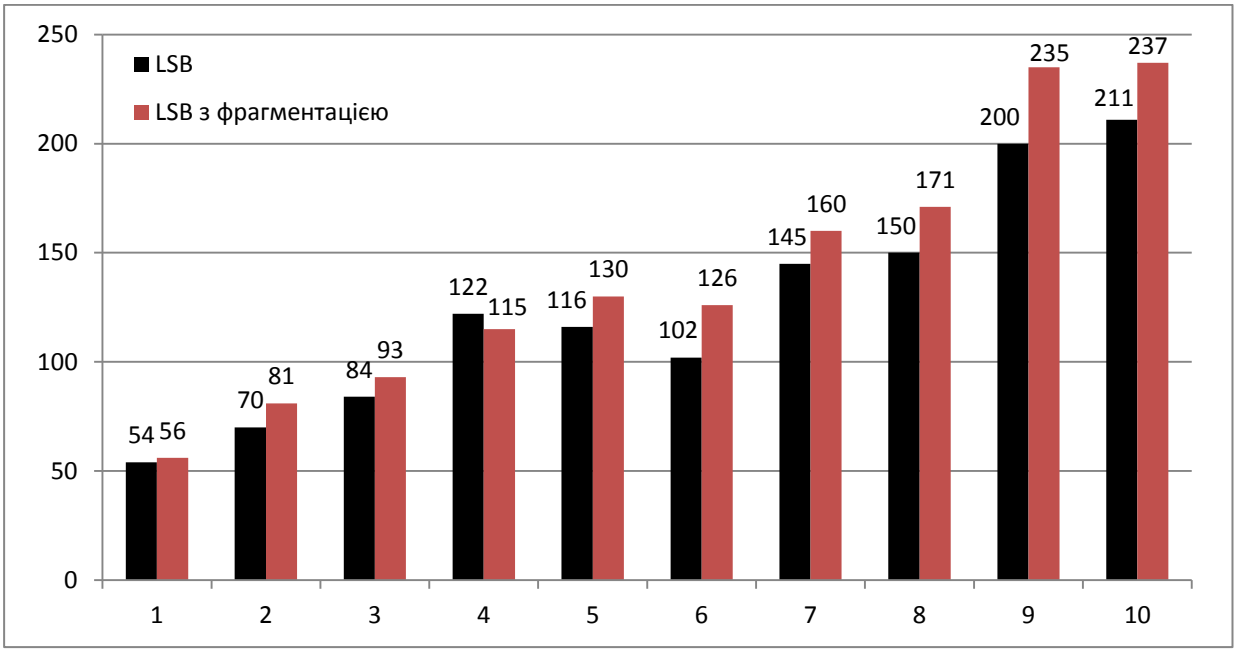

Рис. 1. Порівняння швидкодії *LSB* алгоритму та *LSB* алгоритму з фрагментацією

Варто зазначити, що чим більшим буде обсяг контейнера, тим більший рівень фрагментації стегоданих можна забезпечити за допомогою даної схеми.

Висновки

Запропонований алгоритм передбачає застосування псевдовипадкових чисел для визначення адрес і довжин блоків у поєднанні з алокацією, що забезпечує цілісність даних. Відновлення даних вимагає наявності у користувача закритого ключа, що генерується в процесі фрагментації та може бути збережений та переданий окремими частинами з метою посилення захисту стегоданих. Використання даного алгоритму у традиційній схемі *LSB*-стеганографії зображень призводить лише до незначного збільшення часу обробки даних, що підтверджується експериментальними даними.

Література

1. *Morkel T.* An Overview of Image Steganography [Електронний ресурс] / T. Morkel, J. H. P. Eloff, M. S. Olivier. – Fifth Annual Information Security South Africa Conference (ISSA2005), Sandton, South Africa, June, 2005. – Режим доступу : http://martinolivier.com/open/stegoverview.pdf – [05.03.2012].