ІНФОРМАЦІЙНА БЕЗПЕКА

УДК 004.056.5:003.26:004.415.24

***Н. П. Кухарська**, канд. фіз.-мат. наук, доцент*
(Львівський державний університет безпеки життєдіяльності)

АНАЛІЗ СТЕГАНОГРАФІЧНИХ МЕТОДІВ ДОВІЛЬНОГО ІНТЕРВАЛУ

У статті розглянуто методи текстової стеганографії, а саме, метод зміни інтервалу між реченнями, метод хвостових пропусків, модифікований метод хвостових пропусків. Усі вони належать до когорти методів довільного інтервалу. Їх використовують для організації прихованої передачі конфіденційної інформації відкритими каналами зв'язку. На основі розроблених у середовищі комп'ютерної алгебри MathCAD програмних комплексів покроково відстежено стеганографічні перетворення, що відповідають алгоритмам розглянутих методів. Вивчено питання пропускної здатності побудованих стеганосистем. Вказано на переваги і недоліки кожного методу.

Ключові слова: захист інформації, стеганографія, текстовий контейнер, метод зміни інтервалу між реченнями, метод хвостових пропусків, модифікований метод хвостових пропусків, пропускна здатність, система комп'ютерної алгебри MathCAD.

Н. П. Кухарская

АНАЛИЗ СТЕГАНОГРАФИЧЕСКИХ МЕТОДОВ ПРОИЗВОЛЬНОГО ИНТЕРВАЛА

В статье рассмотрены методы текстовой стеганографии, а именно, метод изменения интервала между предложениями, метод хвостовых пробелов, модифицированный метод хвостовых пробелов. Все они принадлежат к когорте методов произвольного интервала. Их используют для организации скрытой передачи конфиденциальной информации по открытым каналам связи. На основе разработанных в среде компьютерной алгебры MathCAD программных комплексов пошагово отслежено стеганографические преобразования, соответствующие алгоритмам рассмотренных методов. Изучены вопросы пропускной способности построенных стеганосистем. Указано на преимущества и недостатки каждого метода.

Ключевые слова: защита информации, стеганография, текстовый контейнер, метод изменения интервала между предложениями, метод хвостовых пробелов, модифицированный метод хвостовых пробелов, пропускная способность, система компьютерной алгебры MathCAD.

N. P. Kukharska

ANALYSIS OF STEGANOGRAPHY RANDOM INTERVALS METHOD

Text steganography methods, namely, the method of changing the interval between sentences, the method of trailing spaces, modified method of trailing spaces are considered. Completely all of them belong to the cohort of random interval method. They are used for secure data transmission of confidential information in open communication channels. Based on developed application in MathCAD computer algebra software, step by step steganography transformation corresponding considered methods is monitored. The question of the developed steganosystems capacity are studied. The advantages and disadvantages of each methods are specified.

Keywords: information security, steganography, a text container, interval changing between sentences method, method of trailing spaces, modified method of trailing spaces, capacity, MathCAD computer algebra system.

Постановка проблеми. На сучасному етапі розвитку інформаційних систем і технологій, глобальних комп'ютерних систем і засобів мультимедіа як ніколи гостро стоїть питання забезпечення надійності і безпеки даних, що зберігаються у цифровому вигляді, а також відтворення і передачі їх каналами інформаційних комунікацій.

Один із найбільш перспективних і затребуваних на сьогодні підходів до розв'язання цієї проблеми базується на застосуванні методів комп'ютерної стеганографії. Згідно з її алгоритмами, конфіденційні дані, зазвичай невеликого обсягу, приховано вбудовуються у так звані контейнери – будь-які інформаційні масиви і об'єкти цифрового контенту (файли аудіо-, відеоданих, файли текстових форматів, нерухомі зображення і т. ін.), що мають значно більший обсяг і самі по собі не викликають зацікавлення у неавторизованої сторони. Опісля, заповнені контейнери зберігаються у відкритому доступі або передаються легітимному користувачеві незахищеними каналами зв'язку.

Контейнером, як було зауважено вище, можуть бути дані (файли) графічні чи звукові. Простота і велика надлишковість їх структури дають можливість впроваджувати в них додаткову інформацію без особливих труднощів. У зв'язку з цим, стеганографічні методи, які базуються на використанні графічних чи аудіоконтейнерів, набули широкого розповсюдження. Водночас не слід забувати, що переважна більшість цифрової інформації має текстовий вигляд: книги, статті, електронне листування, документи і т.ін. Структура текстових файлів є добре відомою. Тому, виглядає цілком природним їх використання, як контейнерів для стегозахисту під час передавання незахищеними каналами зв'язку повідомлень з обмеженим доступом. Стеганографія, у якій для приховування інформації застосовують текстові контейнери, називається текстовою.

Проблемам розробки і дослідження різних стеганографічних методів присвячені роботи J. Fridrich, G. S. Simmons, T. Filler, L. Perez-Freire, A. Wyner, L. Fearnley, R. Anderson, F. Petitcolas, C. Cachin, В. І. Коржика, І. В. Котенка, Б. Я. Рябка, В. Г. Грібуніна, І. М. Окова, І. В. Туринцева, О. Д. Азарова, М. Є. Шелеста, Г. Ф. Конаховича, А. Ю. Пузиренка, В. О. Хорошка, А. В. Аграновського та інших вчених.

Аналізуючи відомі публікації вітчизняних і закордонних авторів, у яких висвітлені результати досліджень у сфері стеганографічного захисту інформації, доходимо висновку, що питанням текстової стеганографії присвячено порівняно мало робіт.

У Львівському державному університеті безпеки життєдіяльності здійснюється підготовка бакалаврів за спеціальністю "Кібербезпека" (спеціалізація "Управління інформаційною безпекою"). Навчальним планом цієї спеціальності визначено перелік дисциплін, які студент повинен опанувати для досягнення відповідного освітнього ступеня. Серед інших є дисципліна "Основи стеганографії". Відповідно до її навчальної програми курсанти (студенти) мають знати основні алгоритми методів стеганографії, у тому числі текстової, та вміти використовувати їх на практиці для захисту конфіденційних даних. У [1] знаходимо доволі велику кількість прикладів програмних комплексів навчального рівня, створених на основі алгоритмів різних методів. При цьому їх автори задаються ціллю – наочно, крок за кроком продемонструвати весь процес стеганографічного перетворення інформації. Що стосується текстової стеганографії, то тут розглянуто алгоритми лише трьох методів. Метою статті є розширити їх перелік. Окрім запропонованих у [1], детально розглянути теоретичні засади інших відомих методів текстової стеганографії та розробити у середовищі універсальної математичної системи MathCAD навчальні програми, що реалізовуватимуть їх. Сподіваємося, що використаний підхід до викладення матеріалу дасть студентам змогу наочно (на практиці) відстежити алгоритми, закладені в основу методів, пришвидшить процес їх опанування. А це, в свою чергу, сприятиме підвищенню зацікавленості практичними аспектами вирішення завдань із захисту інформації в обчислювальних системах і мережах телекомунікацій.

Кожен стеганографічний метод характеризується пропускною здатністю побудованої на його основі стеганосистеми.

Пропускна здатність – це максимальний обсяг додаткової інформації, що може бути вбудований в один елемент (піксель, відлік, символ) контейнера.

У статті розглянемо методи текстової стеганографії, що належать до так званих методів довільного інтервалу, і порівняємо їх за пропускною здатністю.

Зауважимо, методи довільного інтервалу використовують для приховування даних вільне місце у тексті. Вони оперують інтервалами між реченнями, пропусками в кінці тексто-

вих рядків, інтервалами між словами у тексті, у тому числі, вирівняному за шириною, маніпулюють символами пропуску, що мають різні ASCII-коди. Автори [1] виокремлюють дві обставини, згідно з якими, згадані вище стеганографічні модифікації текстових файлів мали б демонструвати непогані результати. По-перше, зміна кількості пропусків у тексті-контейнері не призводить до зміни значення фраз або речень. По-друге, незаангажований читач навряд чи помітить незначні за розміром текстові прогалини.

Метод зміни інтервалу між реченнями. В основі цього методу лежить така ідея: окремі біти ASCII-кодів (представлених у двійковому форматі) символів конфіденційного повідомлення приховуються у відкритому текстовому файлі після кожного символу кінця речення (це може бути символ крапки у звичайному тексті або символ крапки з комою, наприклад, для програмного коду, написаного мовою C++ тощо) за допомогою одного чи двох пропусків. Одним пропуском стенографічно кодується, наприклад, біт “1”, двома – біт “0”.

Тексти програмних модулів описаного методу з детальним поясненням подані у монографії [1].

Як порожній контейнер розглянемо текст, фрагмент якого наведено на рис. 1. Приховуване повідомлення “Ключ”.

Вільне місце для вбудовування обирається
довільно. Це одночасно перевагою і
недоліком з точки зору прихованості і
даних. Пересічний читач може й не помітити
маніпуляції з текстом. А текстовий редактор
може автоматично змінити кількість і
розмірщення пробілів. Синтаксичні і
семантичні методи взагалі жодним чином не
використовують вільні місця у тексті. Вони
докорінно відрізняються від інших
методів. Проте, всі ці методи можуть
використовуватися одночасно. Тобто,
дублюючи або доповнюючи один одного. Середня
швидкість передавання даних такими методами
становить декілька біт на один кілобайт тексту.

Рисунок 1 – *Фрагмент текстового контейнера, що використовується для приховування даних*

Результат вбудовування першого символу повідомлення (“К”), двійковий формат ASCII-коду якого: $D2B(str2vec("К"))^T = (0_{LSB}\ 1\ 0\ 1\ 0\ 0\ 1\ 1_{MSB})$, зображено на рис. 2. Для наочності і кращого розуміння алгоритму методу зміни інтервалу між реченнями зафарбованими клітинками ми позначили ті пропуски, у яких вбудовано окремі біти конфіденційного повідомлення. Так, у кожному міченому кольором одинарному символі пропуск сховано одиничний біт повідомлення, а у кожній парі – нульовий біт.

Вільне місце для вбудовування обирається
довільно. Це є одночасно перевагою і
недоліком з точки зору прихованості і
даних. Пересічний читач може й не помітити
маніпуляції з текстом. А текстовий редактор
може автоматично змінити кількість і
розмірщення пробілів. Синтаксичні і
семантичні методи взагалі жодним чином не
використовують вільні місця у тексті. Вони
докорінно відрізняються від інших
методів. Проте, всі ці методи можуть
використовуватися одночасно. Тобто,
дублюючи або доповнюючи один одного. Середня
швидкість передавання даних такими методами
становить декілька біт на один кілобайт тексту.

Рисунок 2 – *Фрагмент заповненого методом зміни інтервалу між реченнями текстового контейнера*

Розглянемо недоліки описаного методу. По-перше, він є неефективним, оскільки потребує великого за обсягом тексту порожнього контейнера для вбудовування повідомлення незначного розміру. За умови, що середньостатистичне речення займає 2 рядки по 80 символів кожен, пропускна здатність методу – 0,08%. По-друге, можливість приховування сильно залежить від структури тексту-контейнера (деякі тексти, як наприклад, вільні вірші характеризуються відсутністю стійких узгоджених або однозначних знаків завершення речення). По-третє, деякі з текстових редакторів автоматично додають після знаку кінця речення один-два пропуски (автозавершення), що зводить нанівець процедуру видобування повідомлення.

Метод хвостових пропусків. Метод хвостових пропусків полягає у дописуванні в кінець поточного рядка порожнього файлу-контейнера одного символу пропуску у разі вбудовування в нього, наприклад, нульового біта стеганоповідомлення. Одиничний біт, у той же час, кодуватиметься відсутністю такого символу [2].

Алгоритм методу доволі простий і зрозумілий. Під час вбудовування стеганоповідомлення порожній текстовий файл-контейнер зчитується рядками. При цьому у кінці кожного рядка вилучаються всі “пропускні” символи (пропуски, знаки табуляції, символи повернення каретки і нового рядка). Згодом, залежно від значення поточного біта стеганоповідомлення, поданого у вигляді двійкового вектора, приймається рішення про дописування чи не дописування в кінець рядка символу пропуску. Модифікований рядок записується у файл-результат.

```
Для приховування конфіденційних повідомлень
у тексті (або так звана лінгвістична Стега-
нографія) використовується або звичайна на-
длишковість мови, або формати представлення
тексту. Електронна версія тексту за багать-
ма причинами є найскладнішим місцем для
приховування даних. На відміну від її “жор-
сткої” копії (наприклад, паперової), яка
може бути оброблена як високоструктуроване
зображення і є такою, що легко піддається
різноманітним методам приховування, таким
як незначні зміни формату текстових зразків
регулювання відстані між певними парами си-
мволів (кернінг), відстані між рядками тощо.
В значній мірі це викликане відносним дефі-
цитом у текстовому файлі надлишкової інфор-
мації. Особливо у порівнянні із зображеннями
чи звуковими фрагментами.
```

***Рисунок 3** – Фрагмент текстового контейнера, використаного для приховування даних*

На рис. 4-6 подано послідовність MathCAD-команд, за допомогою яких реалізовується алгоритм методу хвостових пропусків.

```
CC := READBIN("D:\M_TEX7.TXT" , "byte" )

M := "Алгоритм"

D2B(x) := | for i ∈ 1.. 8
          |   | V_i ← mod(x, 2)
          |   | x ← floor(x/2)
          | V
```

***Рисунок 4** – Програмний код – переведення ASCII-коду символа з десяткового формату у двійковий*

```
C :=  i ← 1
      while i ≤ rows(CC)
          C[rows(C)+1] ← CC[i]  if CC[i] ≠ 13
          if CC[i] = 13
              k ← 0
              j ← i
              while CC[j−1] = 32
                  k ← k + 1
                  j ← j − 1
              C ← submatrix(C, 1, rows(C) − k, 1, 1)
              C[rows(C)+1] ← 13
          i ← i + 1
      C
```

Рисунок 5 *– Програмний код – вилучення "пропускних" символів у кінці кожного рядка порожнього контейнера*

```
S :=  Mvec ← str2vec(M)
      Mbin ← D2B(Mvec[1])
      for j ∈ 2..strlen(M)                    if strlen(M) > 1
        Mbin ← stack(Mbin, D2B(Mvec[j]))
      μ ← 1
      Cm ← C
      while μ ≤ 8·strlen(M)
        for i ∈ 1..rows(Cm)
          S[rows(S)+1] ← Cm[i]  if Cm[i] ≠ 13
          if Cm[i] = 13
            Cm ← submatrix(Cm, i + 2, rows(Cm), 1, 1)  if i + 2 ≤ rows(Cm)
            S[rows(S)+1] ← 32  if Mbin[μ] = 0
            μ ← μ + 1
            S[rows(S)+1] ← 13
            S[rows(S)+1] ← 10
            break
      stack(S, Cm)

WRITEBIN("D:\M_T8.TXT", "byte", 1) := S
```

Рисунок 6 *– Програмний код – вбудовування у контейнер даних методом хвостових пропусків*

Під час видобування стеганоповідомлення із заповненого контейнера файл зчитується рядками, а значення поточного біта стеганоповідомлення встановлюється на основі наявності або відсутності символу пропуску в кінці рядка (рис. 7-8).

```
S := READBIN("D:\M_T8.TXT", "byte")

B2D(x) := Σ_{i=1}^{8} (x_i · 2^(i−1))
```

Рисунок 7 *– Програмний код – переведення ASCII-коду з двійкового формату у десятковий*

```
M := | μ ← 1
     | for  i ∈ 1.. rows(S)
     |   if  S_i = 13
     |      | k ← 0
     |      | j ← i
     |      | while  S_{j−1} = 32
     |      |    | k ← k + 1
     |      |    | j ← j − 1
     |      | Mbin_μ ← 1  if  k = 0
     |      | Mbin_μ ← 0  if  k = 1
     |      | μ ← μ + 1
     | for  j ∈ 1.. rows(Mbin) ÷ 8
     |    | kod_sym ← B2D(submatrix(Mbin, 8·j − 7, 8·j, 1, 1))
     |    | break  if  kod_sym = 0
     |    | Mvec_j ← kod_sym
     | Mvec

WRITEBIN("D:\M_T88.TXT" , "byte" , 1) := M
```

Рисунок 8 – *Програмний код – видобування методом хвостових пропусків конфіденційного повідомлення з текстового контейнера*

На рис. 9 подано фрагмент заповненого текстового контейнера, у якому приховано два перші символи повідомлення ”Алгоритм”, а саме: символ ”А”, двійковий формат ASCII-коду якого D2B(str2vec("A"))^T = (0_{LSB} 0 0 0 0 0 1 1_{MSB}) та символ ”л” – D2B(str2vec("л"))^T = (1_{LSB} 1 0 1 0 1 1 1_{MSB}). Для пояснення у дохідливій формі суті алгоритму використано схожий до попереднього методу підхід: у кінці рядків зафарбованими клітинками позначено символи пропуску, що містять нульові біти двійкового вектора секретного повідомлення. Одиничні біти приховують в собі інші рядки, а саме ті, останні символи яких відрізняються від символу пропуск. Наприклад, сьомий-десятий.

Для приховування конфіденційних повідомлень
у тексті (або так звана лінгвістична стега-
нографія) використовується або звичайна на-
длишковість мови, або формати представлення
тексту. Електронна версія тексту за багать-
ма причинами є найскладнішим місцем для
приховування даних. На відміну від її “жор-
сткої” копії (наприклад, паперової), яка
може бути оброблена як високоструктуроване
зображення і є такою, що легко піддається
різноманітним методам приховування, таким
як незначні зміни формату текстових зразків
регулювання відстані між певними парами си-
мволів (кернінг), відстані між рядками тощо.
В значній мірі це викликане відносним дефі-
цитом у текстовому файлі надлишкової інфор-
мації. Особливо у порівнянні із зображеннями
чи звуковими фрагментами.

Рисунок 9 – *Фрагмент заповненого методом хвостових пропусків текстового контейнера*

Переваги методу хвостових пропусків: він може бути застосований до будь-якого тексту; зміни у форматі є досить непомітними, оскільки вільні місця, що використовуються, є периферійними по відношенню до основного тексту.

Пропускна здатність цього методу сильно залежить від використовуваного файлу-контейнера: чим більше у файлі рядків, тим більшу кількість інформації в ньому можна приховати. Якщо прийняти вищеописаний формат файлу-контейнера (кожен рядок має по 80 символів), пропускна здатність побудованої стеганосистеми становитиме 0,15 %. Слід зауважити, що роздрукування файлу-результату призводить до втрати прихованої інформації: на папері або іншому твердому носії неможливо виявити пропуски в кінці рядків. Крім того недоліком цього методу (як і попереднього) є те, що деякі текстові редактори самі по собі додають пропуски в кінець рядків. У той же час, цей метод можна визнати прийнятним для практичного застосування, особливо, якщо використати попереднє стиснення і шифрування стеганоповідомлення.

Модифікований метод хвостових пропусків. Цей метод полягає у дописуванні в кінець кожного рядка текстового контейнера від 0 до 15 символів пропуску, залежно від значення (у десятковому еквіваленті) півбайта ASCII-коду приховуваного символу, поданого у двійковому форматі [3]. Таким чином, для приховування одного символу стеганоповідомлення досить двох рядків файлу-контейнера.

Д	л	я		п	р	и	х	о	в	у	в	а	н	н	я		к	о	н	ф	і	д	е	н	ц	і	й	н	н	х
п	о	в	і	д	о	м	л	е	н	ь		у		т	е	к	с	т	і		(	а	б	о		т	а	к			
з	в	а	н	а		л	і	н	г	в	і	с	т	и	ч	н	а		с	т	е	г	а	н	о	г	р	а	ф	і	я	)
в	и	к	о	р	и	с	т	о	в	у	є	т	ь	с	я		а	б	о		з	в	и	ч	а	й	н	а			
н	а	д	л	и	ш	к	о	в	і	с	т	ь		м	о	в	и	,		а	б	о		ф	о	р	м	а	т	и	
п	р	е	д	с	т	а	в	л	е	н	н	я		т	е	к	с	т	у	.		Е	л	е	к	т	р	о	н	н	а
в	е	р	с	і	я		т	е	к	с	т	у		з	а		б	а	г	а	т	ь	м	а							
п	р	и	ч	и	н	а	м	и		є		н	а	й	с	к	л	а	д	н	і	ш	и	м		м	і	с	ц	е	м
д	л	я		п	р	и	х	о	в	у	в	а	н	н	я		д	а	н	и	х	.									

Рисунок 10 *– Фрагмент оригіналу тексту контейнера, використаного для приховування повідомлення*

Загалом, алгоритм приховування та вилучення стеганоповідомлення не суттєво відрізняється від методу зміни кількості пропусків у кінці текстових рядків. Варіант його реалізації у середовищі MathCAD подано на рис. 11-12.

```
CC := READBIN("D:\M_TEX8.TXT" , "byte" )

M := "Алгоритм"

B2D(x) := Σ_{i=1}^{4} ( x_i · 2^(i−1) )
```

Рисунок 11 – *Програмний код – переведення півбайта ASCII-коду символу з двійкового формату у десятковий*

Далі іде програмний код вилучення “пропускних” символів у кінці кожного рядка порожнього контейнера. Він такий самий як у попередньому методі (рис. 5), як і процедура переведення ASCII-коду символа з десяткового формату у двійковий (рис. 4).

```
S := | Mvec ← str2vec(M)
     | Mbin ← D2B(Mvec_1)
     | for j ∈ 2..strlen(M)                    if strlen(M) > 1
     |   Mbin ← stack(Mbin, D2B(Mvec_j))
     | μ ← 1
     | Cm ← C
     | while μ ≤ 2·strlen(M)
     |   for i ∈ 1..rows(Cm)
     |     S_(rows(S)+1) ← Cm_i   if Cm_i ≠ 13
     |     if Cm_i = 13
     |       Cm ← submatrix(Cm, i + 2, rows(Cm), 1, 1)   if i + 2 ≤ rows(Cm)
     |       ch ← B2D(submatrix(Mbin, 4·μ − 3, 4·μ, 1, 1))
     |       for j ∈ 1..ch          if ch ≠ 0
     |         S_(rows(S)+1) ← 32
     |       μ ← μ + 1
     |       S_(rows(S)+1) ← 13
     |       S_(rows(S)+1) ← 10
     |       break
     | stack(S, Cm)

WRITEBIN("D:\M_T9.TXT","byte",1) := S
```

Рисунок 12 *– Програмний код – вбудовування у контейнер конфіденційного повідомлення модифікованим методом хвостових пропусків*

Видобути стеганоповідомлення з заповненого контейнера можна так, як показано на рис. 13.

```
S := READBIN("D:\M_T9.TXT","byte")

B2D(x) := Σ_(i=1)^8 (x_i · 2^(i−1))

D2B(x) := | for i ∈ 1..4
          |   V_i ← mod(x, 2)
          |   x ← floor(x/2)
          | V
```

```
M := | μ ← 1
     | for i ∈ 1..rows(S)
     |   if S_i = 13
     |     | k ← 0
     |     | j ← i
     |     | while S_(j−1) = 32
     |     |   | k ← k + 1
     |     |   | j ← j − 1
     |     | Mbin1 ← D2B(k)
     |     | Mbin ← Mbin1  if μ = 1
     |     | Mbin ← stack(Mbin, Mbin1)  if μ > 1
     |     | μ ← μ + 1
     | for j ∈ 1..rows(Mbin) ÷ 8
     |   | kod_sym ← B2D(submatrix(Mbin, 8·j − 7, 8·j, 1, 1))
     |   | break  if kod_sym = 0
     |   | Mvec_j ← kod_sym
     | Mvec
WRITEBIN("D:\M_T0.TXT", "byte", 1) := M
```

***Рисунок 13** – Програмний код – видобування модифікованим методом хвостових пропусків конфіденційного повідомлення з текстового контейнера*

На рис. 14 подано результат вбудовування у текстовий контейнер перших чотирьох символів повідомлення “Алгоритм”. Нагадаємо, двійкове подання ASCII-коду символу “А” таке: $0_{LSB}\ 0\ 0\ 0\ 0\ 0\ 1\ 1_{MSB}$. Розіб'ємо його на два півбайти: $0_{LSB}\ 0\ 0\ 0_{MSB}$ та $0_{LSB}\ 0\ 1\ 1_{MSB}$, десяткове значення яких 0 і 12 відповідно. Порівнюючи рис. 10 та рис. 14, бачимо, що у кінці другого рядка (рис. 14) з’явилося дванадцять пропусків, а в кінці першого рядка вони відсутні. Такий стан справа повністю узгоджується з попередніми міркуваннями. Щоб у текстовому контейнері (рис. 10) приховати символ ”А”, слід дописати нуль та дванадцять символів “пропуск” в кінці першого та другого рядка відповідно. Аналогічно, $D2B(str2vec("л"))^{T} = (1_{LSB}\ 1\ 0\ 1\ 0\ 1\ 1\ 1_{MSB})$ складається з двох півбайтів $1_{LSB}\ 1\ 0\ 1_{MSB}$, $0_{LSB}\ 1\ 1\ 1_{MSB}$, у десятковому форматі значення яких 11 та 14. У зв’язку з цим, саме таку кількість символів “пропуск” (одинадцять і чотирнадцять) маємо у третьому та четвертому рядку файла-результату (рис. 14). Їх функція – приховувати символ “л” повідомлення.

Переваги та недоліки модифікованого методу хвостових пропусків ті ж, що і в попередньому випадку. Пропускна здатність – 0,63 %. Збільшення у порівняні з методом хвостових пропусків пропускної здатності призводить до того, що стеганограму можна виявити легше.

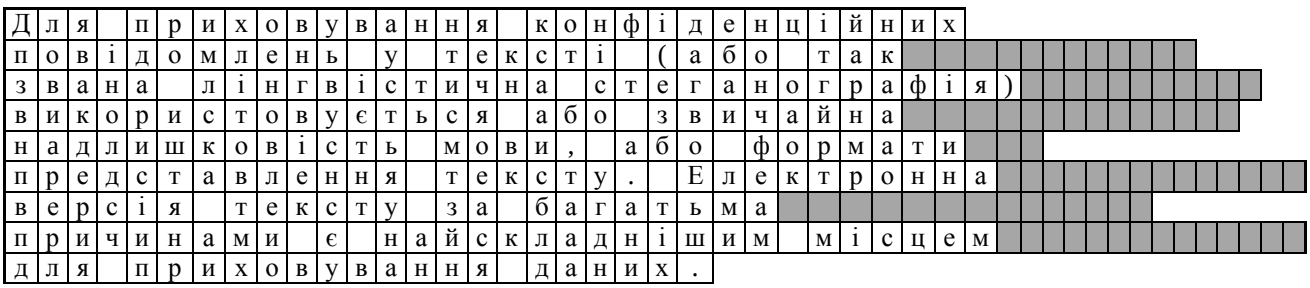

***Рисунок 14 –** Фрагмент заповненого модифікованим методом хвостових пропусків текстового контейнера*

У таблицю 1 зведемо відомості, що стосуються пропускної здатності розглянутих методів та можливості видобування прихованої інформації з паперових носіїв.

Таблиця 1

Порівняльний аналіз методів щодо пропускної здатності та можливості видобування прихованої інформації з паперових носіїв

№ з/п	Назва методу	Пропускна здатність	Можливість видобування прихованої інформації з паперових носіїв
1.	Метод зміни інтервалу між реченнями	0,08%	+
2.	Метод хвостових пропусків	0,15%	–
3.	Модифікований метод хвостових пропусків	0,63%	–

Висновок. У статті розглянуто стеганографічні методи, алгоритми яких базуються на маніпуляціях з символом “пропуск”. Вони належать до методів текстової стеганографії, а точніше до методів довільного інтервалу і використовуються для вбудовування у текстові файли конфіденційних повідомлень з метою їх замаскованої передачі відкритими каналами комп’ютерних мереж.

Систематизований та поданий у статті матеріал може бути використаний також у навчальних цілях для підготовки фахівців з програмного захисту інформації.

Список літератури:

1. Конахович Г. Ф. Компьютерная стеганография. Теория и практика / Г. Ф. Конахович, А. Ю. Пузыренко. – К. : МК-Пресс, 2006. – 249 с.

2. Иванов В. Текстовая стеганография: метод хвостовых пробелов [Электронный ресурс] / Иванов В. – 2012. – Режим доступа : http://www.nestego.ru/2012/05/blog-post_03.html#more

3. Иванов В. Текстовая стеганография: модифицированный метод хвостовых пробелов [Электронный ресурс] / Иванов В. – 2012. – Режим доступа : http://www.nestego.ru/2012/05/blog-post_04.html#more

References:

1. Konakhovych, G. F. and Puzyrenko, A. Yu. (2006). Computer steganography. – Kyiv: MK-PRESS (in Russ.)

2. Ivanov, V. Text steganography: the method of trailing spaces. Retrieved from http://www.nestego.ru/2012/05/blog-post_03.html#more

3. Ivanov, V. (2012) Text Steganography: modified method of trailing spaces. Retrieved from http://www.nestego.ru/2012/05/blog-post_04.html#more

❖❖❖