Вісник ЛДУБЖД

Bulletin of Lviv State University of Life Safety

ISSN 2078-4643
https://journal.ldubgd.edu.ua/index.php/Visnuk/
УДК 004.056.5:003.26:004.415.24
DOI: 10.32447/20784643.18.2018.03

Н. П. Кухарська
Львівський державний університет безпеки життєдіяльності

ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ ПРИХОВУВАННЯ ІНФОРМАЦІЇ МЕТОДАМИ ДОВІЛЬНОГО ІНТЕРВАЛУ

На сучасному етапі розвитку інформаційних систем і технологій, глобальних комп'ютерних систем і засобів мультимедіа як ніколи гостро стоїть питання забезпечення надійності, безпеки зберігання цифрових даних та передачі їх відкритими каналами інформаційних комунікацій.

Один із найбільш перспективних і затребуваних на сьогодні підходів до розв'язання цієї проблеми базується на застосуванні методів комп'ютерної стеганографії.

Метою статті є систематизувати відомості щодо методів текстової стеганографії, а саме методів довільного інтервалу, здійснити порівняльний їх аналіз за пропускною здатністю.

Методи дослідження – методи текстової стеганографії: метод подвійних пропусків між словами, метод зміни коду пропуску, метод зміни кількості пропусків у кінці текстових рядків, метод зміни кількості пропусків між словами вирівняного за шириною тексту.

Методи довільного інтервалу використовують для приховування даних вільне місце у тексті. Вони оперують інтервалами між реченнями, пропусками в кінці текстових рядків, інтервалами між словами у тексті, у тому числі, вирівняному за шириною, маніпулюють символами пропуску, що мають різні ASCII-коди. Їх використовують для організації прихованого передавання конфіденційної інформації відкритими каналами зв'язку.

У статті на основі розроблених у середовищі комп'ютерної алгебри MathCAD програмних комплексів, покроково відстежено стеганографічні перетворення, що відповідають алгоритмам методів.

Вивчено питання пропускної здатності побудованих стеганосистем.

Пропускна здатність – це максимальний обсяг додаткової інформації, що може бути вбудований в один елемент (символ) текстового контейнера. Так, пропускна здатність методу подвійних пропусків між словами та методу зміни коду пропуску у випадку українськомовного текстового контейнера – 1,75 %. Метод зміни кількості пропусків між словами вирівняного за шириною тексту має нижчу пропускну здатність – 0,4 %. Пропускна здатність методу зміни кількості пропусків у кінці текстових рядків залежить від різниці між кількістю символів у найдовшому рядку та усіма іншими рядками.

У статті також вказано на переваги і недоліки кожного методу.

Висновки. Методи довільного інтервалу є ефективними за умови, що текст представлено у форматі ASCII. Загалом, текстові файли є "незручними" контейнерами. Їм бракує надлишковості у порівняні, наприклад, з графічними чи аудіофайлами.

Для таких методів довільного інтервалу як метод зміни коду пропуску, метод зміни кількості пропусків у кінці текстових рядків характерним є те, що приховані дані неможливо отримати з твердої копії текстового файлу.

Незважаючи на недоліки, методи довільного інтервалу мають підстави бути застосованими через розповсюдженість файлів текстового формату. Користувачі комп'ютерних мереж постійно обмінюються текстовими повідомленнями. Це є звичною повсякденною справою, тому текстові файли, навіть такі, що містять приховану конфіденційну інформацію, не мали би викликати зайву цікавість у сторонніх осіб.

Ключові слова: захист інформації, стеганографія, текстовий контейнер, метод подвійних пропусків між словами, метод зміни коду пропуску, метод зміни кількості пропусків у кінці текстових рядків, система комп'ютерної алгебри MathCAD.

Цифрова стеганографія – відносно молода галузь знань, початок розвитку якої припадає на 90-і роки XX століття. Застосовувані цифровою стеганографією підходи сьогодні становлять інтерес як для фахівців у сфері захисту інформації, так і для дослідників, що вивчають питання теорії інформації, цифрової обробки сигналів.

Провівши аналіз публікацій, серед широкого загалу присвячених стеганографії робіт виокремимо монографії [1-3] як базові за часом їх видання, цитованістю в інших джерелах, обсягом представленого матеріалу, а також за кількістю опрацьованих і систематизованих їхніми авторами наукових праць з обраної проблематики. У той же час, стеганографічна наука не стоїть на місці, вона постійно розвивається, про що свідчать численні публікації та захисти дисерта-

цій, конференції, патенти на винаходи, свідоцтва про реєстрацію програмного забезпечення.

Метою цієї статті є систематизувати відомості щодо методів текстової стеганографії, а саме методів довільного інтервалу, розібратись з алгоритмами, реалізувавши їх у середовищі універсальної математичної системи MathCAD. Ця стаття є продовженням попередньої [4], у якій ми ділилися досвідом викладання у Львівському державному університеті безпеки життєдіяльності дисципліни "Основи стеганографії", вивчення якої передбачено навчальною програмою підготовки бакалаврів за спеціальністю "Кібербезпека". У статті [4] детально описано три методи текстової стеганографії, що належать до групи методів довільного інтервалу, а саме: метод зміни інтервалу між реченнями, метод хвостових пропусків, метод модифікованих хвостових пропусків. Продовжуючи тему, зберігаючи стиль подання матеріалу [4], розглянемо решта методів цієї групи та порівняємо їх за пропускною здатністю.

Метод подвійних пропусків між словами. Згідно з алгоритмом методу, стеганографічне перетворення текстового контейнера здійснюється у такий спосіб:

1. Спершу у тексті-контейнері вилучають всі зайві пропуски між словами. У результаті отримують текст, у якому слова відокремлюються між собою лише одним символом "пропуск".

2. Далі, перетворене у двійкову послідовність конфіденційне повідомлення по одному біту вбудовують у підготовлений описаним вище чином текст за таким принципом: якщо приховують нульовий біт, то між словами залишають один символ пропуску, якщо приховують одиничний біт, то до наявного у тексті пропуску додають ще один.

На рис. 1-3 подано послідовність MathCAD-команд, за допомогою яких реалізовується алгоритм методу подвійних пропусків між словами.

```
CC := READBIN("D:\M-TEXT7.TXT","byte")

M := "Algorithm"

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

Рисунок 1 – Програмний код – переведення ASCII-коду символа з десяткового формату у двійковий

```
C := | i ← 1
     | while i ≤ rows(CC)
     |   C_(rows(C)+1) ← CC_i
     |   i ← i + 1
     |   while CC_i = 32 if (i ≤ rows(CC)) ∧ (CC_(i−1) = 32)
     |     i ← i + 1
     | C
```

Рисунок 2 – Програмний код – вилучення зайвих між словами символів пропуску

```
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
     | i ← 1
     | while μ ≤ 8·strlen(M) ∧ (i < rows(C))
     |   S_(rows(S)+1) ← C_i
     |   if C_i = 32
     |     S_(rows(S)+1) ← 32 if Mbin_μ = 1
     |     μ ← μ + 1
     |   i ← i + 1
     | S ← stack(S, submatrix(C, i, rows(C), 1, 1)) if i < rows(C)
     | S

WRITEBIN("D:\M-TEXT10.TXT","byte",1) := S
```

Рисунок 3 – Програмний код – вбудовування у текстовий контейнер даних методом подвійних пропусків між словами

Розглянемо текст [3], фрагмент якого наведено на рис. 4, у ролі порожнього контейнера.

Результат вбудовування у цей контейнер повідомлення "Algorithm" зображено на рис. 5. ASCII-коди перших двох символів повідомлення "A" та "l" у двійковому форматі мають такий вигляд:

$D2B(str2vec("A"))^T = (1_{LSB}\ 0\ 0\ 0\ 0\ 0\ 1\ 0_{MSB})$,
$D2B(str2vec("l"))^T = (0_{LSB}\ 0\ 1\ 1\ 0\ 1\ 1\ 0_{MSB})$.

На рис. 5 сірим кольором виділено символи пропуску, що приховують конфіденційне повідомлення. Кожній парі зафарбованих клітинок з символами "пропуск" відповідає одиничний біт вбудованого у контейнер повідомлення, а кожній зафарбованій одинарній клітинці – нульовий біт.

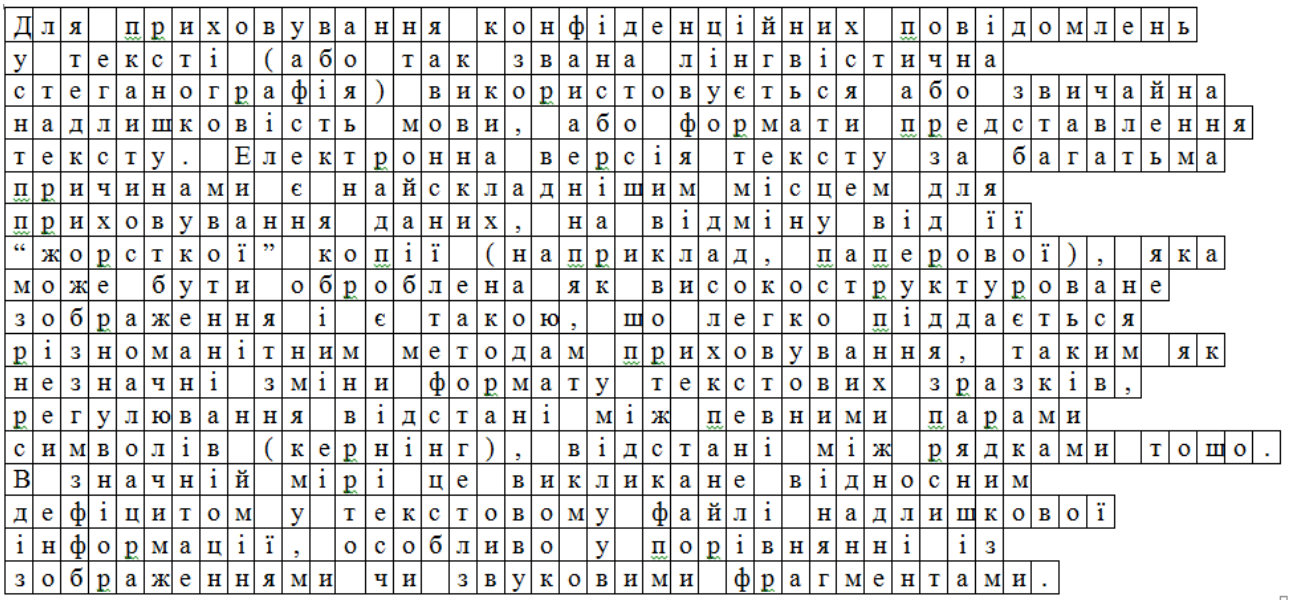

Рисунок 4 – Фрагмент тексту порожнього контейнера

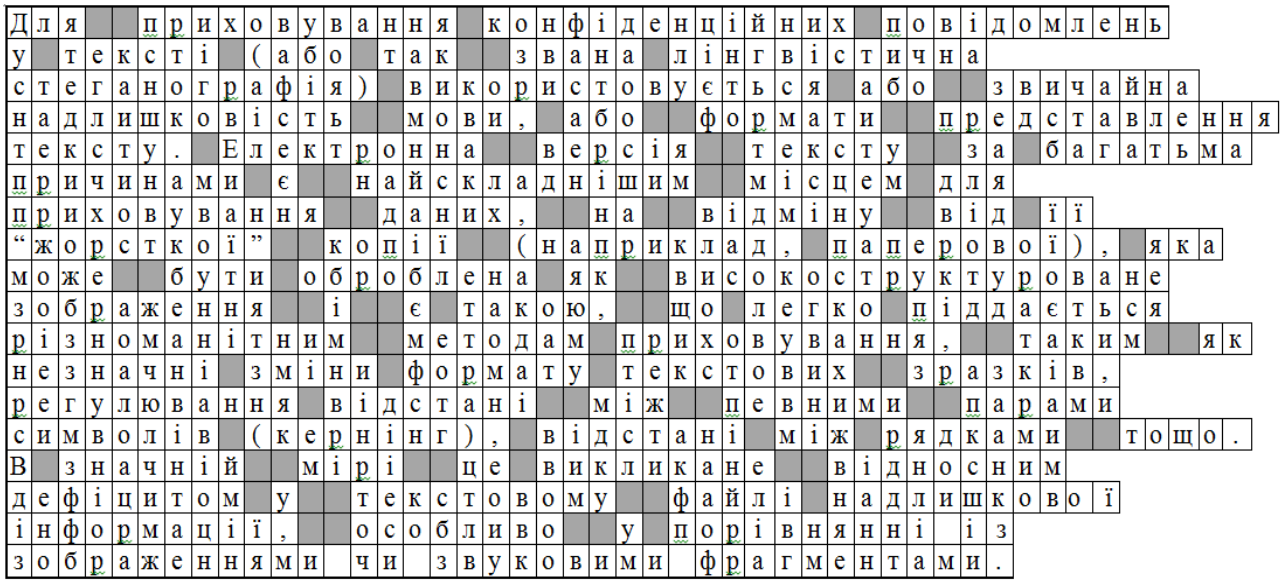

Рисунок 5 – Фрагмент текстового контейнера заповненого методом подвійних пропусків

Видобування прихованого повідомлення здійснюється за допомогою програмних модулів, поданих на рис. 6-7. Критеріями зупинки процесу видобування є: досягнутий кінець файлу-контейнера або нульове значення ASCII-коду чергового видобутого символу.

$$S := \text{READBIN}(\text{"D:\M-TEXT10.TXT"}, \text{"byte"})$$

$$B2D(x) := \sum_{i=1}^{8} \left(x_i \cdot 2^{i-1} \right)$$

Рисунок 6 – Програмний код – переведення ASCII-коду з двійкового формату у десятковий

```
M := | μ ← 1
     | j ← 1
     | i ← 1
     | while i < rows(S)
     |     if S_i = 32
     |         Mbin_μ ← 0 if S_(i+1) ≠ 32
     |         if S_(i+1) = 32
     |             Mbin_μ ← 1
     |             i ← i + 1
     |         μ ← μ + 1
     |     if μ = 9
     |         dop_z ← B2D(Mbin)
     |         break if dop_z = 0
     |         Mvec_j ← dop_z
     |         j ← j + 1
     |         μ ← 1
     |     i ← i + 1
     | Mvec

WRITEBIN("D:\M-TEXT1.TXT","byte",1) := M
```

Рисунок 7 – Програмний код – видобування з текстового контейнера конфіденційного повідомлення методом подвійних пропусків між словами

Оцінимо пропускну здатність [4] побудованої стеганосистеми. Для цього скористаємось результатами досліджень частоти повторюваності букв та символу пропуск між словами у відкритих текстах українською мовою [5]. Середньостатистична відносна частота символу "пропуск" – 0,138. Це означає, що у будь-якому випадково обраному фрагменті українськомовного тексту обсягом сто символів символ пропуску зустрінеться приблизно 14 разів. Щоб приховати один символ конфіденційних даних (8 біт), потрібно у тексті-контейнері знайти, згідно з алгоритмом методу, 8 пропусків. Виходячи із наведених вище відомостей, таку кількість мав би містити текст із 57 символів. Обчислимо пропускну здатність методу: $\frac{1}{57}\times 100\% = 1,75\%$. Це хороший результат.

До переваг розглянутого методу слід віднести простоту і прозорість програмної реалізації. Зауважимо, наявність у тексті подвійних пропусків не завжди свідчить про те, що у тексті міститься приховане повідомлення.

Метод зміни коду пропуску. У багатьох кодових сторінках символи, які відображаються у тексті як прогалина, кодуються різними кодами. Так, у кодовій сторінці Windows-1251 звичайний пропуск має код 32, а, так званий, нерозривний пропуск – код 160.

Стеганоповідомлення вбудовується у символи пропуску текстового файла згідно з алгоритмом методу таким чином: біт "1" кодується символом нерозривного пропуску, а біт "0" – символом звичайного пропуску [6]. Якщо у контейнері черговий символ пропуску має код 32 і в ньому слід приховати біт "1", то його замінюють на пропуск з кодом 160. І, навпаки, приховуючи біт "0", прослідковують, щоб він був таким, який має код 32, інакше – здійснюють відповідну заміну.

Нижче подано програмні коди (рис. 8)

приховування стеганоповідомлення на основі методу зміни коду пропуску.

```
M := "Algorithm"
Cm := READBIN("D:\M_TEX7.TXT","byte")
```

Далі має бути програмний код переведення ASCII-коду символа з десяткового формату у двійковий (рис. 6).

Як порожній контейнер розглянемо той

```
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
     | for i ∈ 1..rows(Cm)
     |     if μ > 8·strlen(M)
     |         S_(rows(S)+1) ← Cm_i if Cm_i ≠ 160
     |         S_(rows(S)+1) ← 32 if Cm_i = 160
     |     if μ ≤ 8·strlen(M)
     |         S_(rows(S)+1) ← Cm_i if Cm_i ≠ 32 ∧ Cm_i ≠ 160
     |         if Cm_i = 32 ∨ Cm_i = 160
     |             S_(rows(S)+1) ← 32 if Mbin_μ = 0
     |             S_(rows(S)+1) ← 160 if Mbin_μ = 1
     |             μ ← μ + 1
     | S
```

самий текст, що і у попередньому методі (рис. 4).

```
WRITEBIN("D:\M_M55.TXT","byte",1) := S
```

Рисунок 8 – Програмний код – вбудовування у контейнер конфіденційного повідомлення методом зміни коду пропуску

Результат вбудовування повідомлення "Algorithm" подано на рис. 9. Зафарбовані клітинки означають:

– символ пропуску з ASCII-кодом 160, у який вбудовано одиничний біт;

– символ пропуску з ASCII-кодом 32, у який вбудовано нульовий біт.

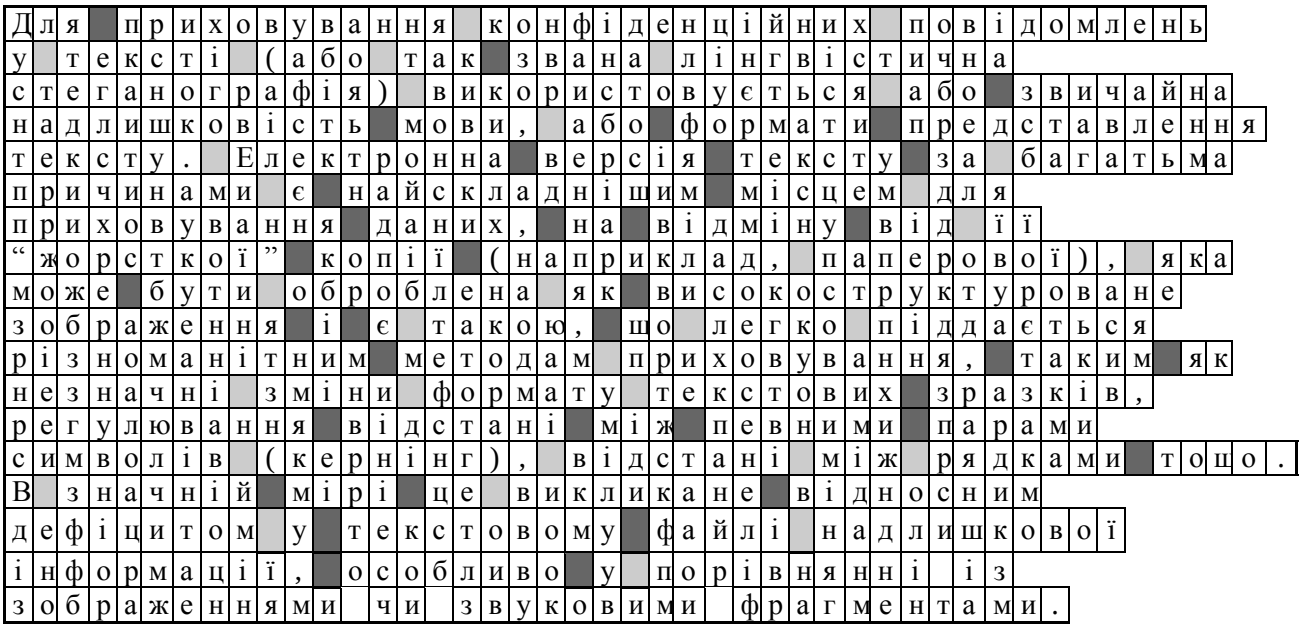

Рисунок 9 – Фрагмент текстового контейнера заповненого методом зміни коду пропуску

Видобути стеганоповідомлення з заповненого контейнера можна так, як показано на рис. 10.

```
S := READBIN("D:\M_M55.TXT","byte")

B2D(x) := Σ_{i=1}^{8} (x_i · 2^(i−1))

M := μ ← 1
     j ← 1
     for i ∈ 1..rows(S)
       if S_i = 32
         Mbin_μ ← 0
         μ ← μ + 1
       if S_i = 160
         Mbin_μ ← 1
         μ ← μ + 1
       if μ = 9
         dop_z ← B2D(Mbin)
         break if dop_z = 0
         Mvec_j ← dop_z
         j ← j + 1
         μ ← 1
     Mvec

WRITEBIN("D:\M_M6.TXT","byte",1) := M
```

Рисунок 10 – Програмний код – видобування з текстового контейнера конфіденційного повідомлення методом зміни коду пропусків

Текст з різними кодами пропусків без проблем відображається у всіх перевірених нами текстових редакторах і засобах перегляду.

Пропускну здатність методу зміни коду пропусків обчислюємо у такий самий спосіб як і для попереднього методу. Вона дорівнює 1,75 %.

Метод зміни кількості пропусків у кінці текстових рядків. Цей метод також належить до методів довільного інтервалу, позаяк для вбудовування конфіденційних даних він використовує вільні місця у тексті і полягає у дописуванні різної кількості пропусків у кінець текстових рядків. Кількість пропусків, яку слід додати до того чи іншого рядка, обчислюють як різницю між кількістю символів у найдовшому рядку тексту і кількістю символів у поточному рядку. Тип кожного доданого пропуску залежить від значення біта, що вбудовується. Якщо приховується біт “0”, то до стеганограми дописується звичайний пропуск (ASCII-код – 32), а якщо біт “1” – нерозривний пропуск (ASCII-код – 160).

Тексти програмних модулів цього методу з детальним їх поясненням подані у монографії [3].

На рис. 11 зображено фрагмент тексту (рис. 4), у який вбудовано повідомлення “Algorithm” методом зміни кількості пропусків у кінці текстових рядків. Тут використано ті ж самі позначення, що і у попередньому методі.

Недоліком розглянутого методу є те, що деякі програми можуть ненавмисно видаляти хвостові пропуски.

Пропускна здатність стеганосистеми залежить від різниці між кількістю символів у найдовшому рядку та усіма іншими.

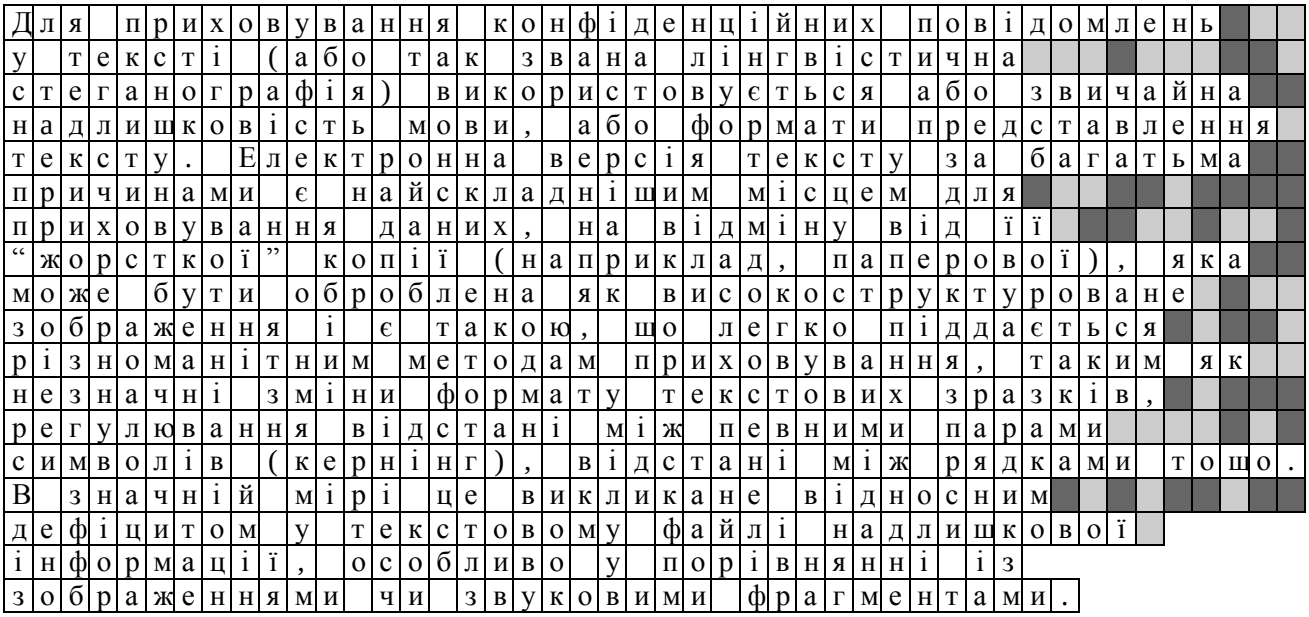

Рисунок 11 – Фрагмент текстового контейнера заповненого методом зміни кількості пропусків у кінці текстових рядків

Метод зміни кількості пропусків між словами вирівняного за шириною тексту. Відповідно до алгоритму методу, конфіденційне повідомлення приховують у символах пропуску контейнера, у той же час надаючи текстові вигляду вирівняного за шириною. Як і у попередніх методах, секретні дані представляють у вигляді бітової послідовності. У процесі вбудовування розглядають лише ті слова порожнього контейнера, які оточені пропусками з обох боків. У залежності від значення приховуваного біта додають додатковий символ ”пропуск” перед словом або після нього. Якщо у текст вбудовується біт “0”, то символ пропуску додають після слова, якщо “1”, то перед словом. Описаним вище чином до одного рядка можна вбудувати декілька біт. Щоправда, у зв’язку з необхідністю вирівнювати текст за шириною, не кожну пару пропусків, що оточують слово, можна використати для приховування даних. Під час видобування конфіденційної інформації пари одинарних та пари подвійних пропусків заповненого контейнера інтерпретують як такі, що не містять прихованої інформації.

Коди програм, що реалізовують алгоритм методу зміни кількості пропусків між словами вирівняного за шириною тексту, подано у монографії [3].

Як порожній контейнер розглянемо текст (рис. 12). Стеганоповідомлення – “Algorithm”.

На рис. 13 зображено фрагмент текстового контейнера, заповненого методом зміни кількості пропусків між словами вирівняного за шириною тексту.

Три розташовані підряд клітинки вигляду ■ вказують на місце приховання у текстовому контейнері одного біта зі значенням “1”, а тріада клітинок ▪ – на місце приховання одного біта зі значенням “0”.

Як бачимо, у першому рядку текстового контейнера (рис. 13) приховано лише один одиничний біт, у другому рядку – два нульових біти, у третьому – один нульовий, у всьому фрагменті тексту – десять біт стеганоповідомлення.

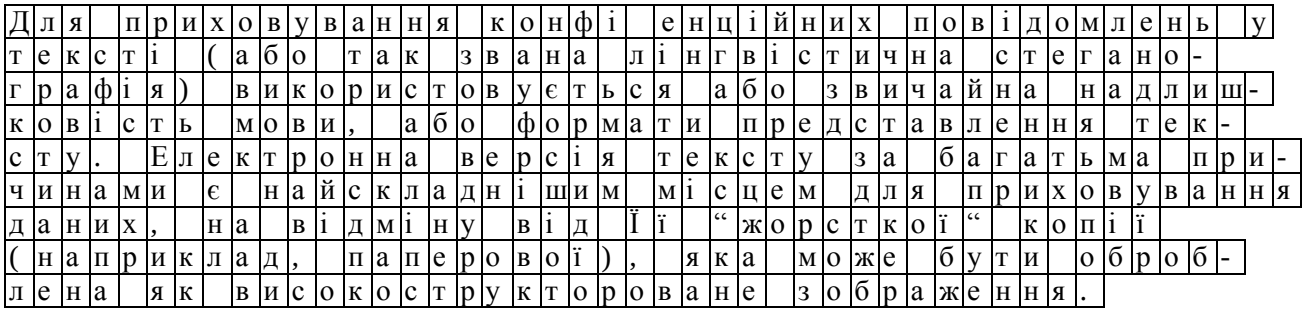

Рисунок 12 – Фрагмент тексту порожнього контейнера

Рисунок 13 – Фрагмент текстового контейнера, заповненого методом зміни кількості пропусків між словами вирівняного за шириною тексту

Пропускна здатність цієї стеганосистеми невисока, вона є меншою за 0,4 %.

Для кращого сприйняття матеріалу відомості, які стосуються пропускної здатності розглянутих методів та можливості видобування прихованої інформації з паперових носіїв, подамо у вигляді табл. 1.

Таблиця 1

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

№ з/п	Метод довільного інтервалу	Пропускна здатність	Можливість видобування прихованої інформації з паперових носіїв
1.	Метод подвійних пропусків між словами	1,75 %	+
2.	Метод зміни коду пропуску	1,75%	–
3.	Метод зміни кількості пропусків у кінці текстових рядків	Залежить від різниці між кількістю символів у найдовшому рядку та всіма іншими	–
4.	Метод зміни кількості пропусків між словами вирівняного за шириною тексту	0,4%	+

Висновок. Розглянуті методи довільного інтервалу є ефективними за умови, що текст представлено у форматі ASCII. Загалом, текстові файли є “незручними” контейнерами. Їм бракує надлишковості у порівнянні, наприклад, з графічними чи аудіофайлами.

Як зазначалося вище і у статті [4], для деяких методів довільного інтервалу характерним є те, що приховані дані неможливо отримати з твердої копії текстового файлу. Друковані документи висувають до процесу приховування вимоги, які виходять за рамки можливостей текстового файлу при кодуванні ASCII. Щоправда, і у “жорстких” копіях тексту можна приховувати повідомлення, використовуючи при цьому інші підходи. Наприклад, можна незначно змінювати відстані між певними парами символів, позиції базових ліній (ліній, на яких лежать найнижчі елементи літер, знаки пунктуації) чи накреслення символів.

Список літератури:

1) Аграновский А. В., Балакин А. В., Грибунин В. Г., Сапожников С. А. Стеганография, цифровые водяные знаки и стегоанализ : монография. Москва : Вузовская книга, 2009. 220 с.

2) Грибунин В. Г., Оков И. Н., Туринцев И. В. Цифровая стеганография. Москва : Солон-Пресс, 2009. 272 с.

3) Конахович Г. Ф., Пузыренко А. Ю. Компьютерная стеганография. Теория и практика. Киев : МК-Пресс, 2006. 249 с.

4) Кухарська Н. П. Аналіз стеганографічних методів довільного інтервалу. *Вісник ЛДУ БЖД*. Львів, 2016. № 14. С. 7-16.

5) Иванов В. Текстовая стеганография: метод двойных пробелов между словами. URL: http://www.nestego.ru/2012/05/blog-post_12.html.

6) Сушко С.О., Фомичова Л.Я., Барсуков Є.С. Частоти повторюваності букв і біграм у відкритих текстах українською мовою. *Захист інформації*. Київ, 2010. Т. 12, № 3. С. 94-102.

7) Иванов В. Текстовая стеганография: метод изменения кода пробела. URL: http://www.nestego.ru/2012/05/blog-post_11.html.

References:

1) Gribunin, V.G., Okov, I.N. and Turintsev, I. V. (2009). *Digital steganography*. Moscow: Solon-Press (in Russ.)

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

3) Agranovskiy, A. V., Balakin, A. V., Gribunin, V. G. and Sapozhnikov, S. A. (2009).

Steganography, digital watermarks and stegoanalysis. Moscow: Vuzovskaia kniga (in Russ.)

4) Kukharska N. P. (2016). Analysis of steganography methods of random interval. *Bulletin of Lviv State University of Life Safety, 14,* 7-16 (in Ukr.)

5) Ivanov, V. (2012). Text *steganography: the method of double spaces between word.* Retrieved from http://www.nestego.ru/2012/05/blog-post_12.html (in Russ.)

6) Sushko, S. O., Fomychova, L. Ya. and Barsukov Ye. S. (2010). Frequencies of repetition of letters and bigram in open texts in Ukrainian. *Zahist ìnformacìï (Ukrainian Information Security Research Journal), 12, 3,* 94-102. (in Ukr.)

7) Ivanov, V. (2012). *Text steganography: the method of changing the space code.* Retrieved from http://www.nestego.ru/2012/05/blog-post_11.html (in Russ.)

N. P. Kukharska

PROGRAM IMPLEMENTATION OF ALGORITHMS OF HIDING THE INFORMATION BY METHODS OF A RANDOM INTERVAL

At the current stage of the development of information systems and technologies, global computer systems and multimedia tools, as never before, there is an urgent need to ensure the reliability, security of storage of digital data and their transmission through open channels of information communications.

One of the most promising and popular approaches to solving this problem is based on the applying of computer steganography methods.

The purpose of the article is systematizing information about methods of textual steganography, namely methods of arbitrary intervals, carrying out a comparative bandwidth analysis.

Methods of research - methods of textual steganography: the method of double spaces between words, the method of changing the space code, the method of changing the number of spaces at the end of text strings, the method of changing the number of spaces between words aligned to the width of the text.

Methods of arbitrary interval are used to hide the data in the free space in the text. They use intervals between sentences, spaces at the end of text strings, intervals between words in the text, including those, which are aligned in width, manipulate symbols of spaces, which have different ASCII codes. They are used to organize secret transmission of confidential information through open communication channels.

In the article, on the basis of the software complexes developed in the environment of the computer algebra MathCAD, the steganographic transformations, which consistent with the algorithms of the methods, are sequentially tracked.

The question of the bandwidth of the constructed steganosystems was considered.

Bandwidth is the maximum amount of additional information that can be embedded in one element (symbol) of the text container.

So, the bandwidth of the double-space method between words and the method of changing the space code in the case of the Ukrainian-language text container is 1.75%. The method of changing the number of spaces between aligned by the width of the text of the words has a lower bandwidth of 0.4%. The bandwidth of the method of changing the number of spaces at the end of the text strings depends on the difference between the number of symbols in the longest line and all other lines.

The article also outlines the advantages and disadvantages of each method.

Conclusions. The arbitrary interval methods are effective provided that the text is presented in ASCII format. In general, text files are "inconvenient" containers. They lack redundancy in comparison, for example, with graphic or audio files.

For such arbitrary interval methods as a method of changing the space code, the method of changing the number of spaces at the end of the text strings is characterized by the fact that the hidden data can not be obtained from a hard copy of the text file.

Despite the drawbacks, arbitrary interval methods have reason to be applied because of the prevalence of text file files. Users of computer networks are constantly exchanging text messages. This is a routine everyday action, so text files, even those, which contain hidden confidential information, should not cause unnecessary interest among outsiders.

Keywords: information protection, steganography, text container, double spaces method between words, method for changing the code of spacing, method for changing the number of spaces at the end of text strings, the system of computer algebra MathCAD.