УДК 004.056.55

К.Ю. ШЕХАНІН, А.О. КОЛГАТІН, Є.Є. ДЕМЕНКО, О.О. КУЗНЕЦОВ, д-р техн. наук

ПРИХОВУВАННЯ ДАНИХ У СТРУКТУРУ ФАЙЛОВОЇ СИСТЕМИ СІМЕЙСТВА FAT

Вступ

Інформаційні технології визначають процеси передачі і розповсюдження, зберігання та обробки інформації, а також її використання у певних цілях [1 – 3]. Інколи факт виконання цих процесів повинен бути прихований від сторонніх осіб. Цим і займається галузь науки цифрова стеганографія [4 – 8].

Окремим розділом сучасної цифрової стеганографії, що вивчає методи і засоби вбудовування та вилучення інформаційних повідомлень в різні цифрові контейнери з використанням технічних особливостей зберігання, передачі і відображення даних, є технічна стеганографія. Вона вивчає такі методи, застосована надмірність в яких штучна і її поява зумовлена технічними особливостями обробки та передачі цифрових контейнерів [9 – 14].

На даний час відомо декілька напрямків розвитку технічної стеганографії: прихована передача інформації у мережевому трафіку [9]; приховування інформації у модель під час 3D-друку [10]; методи технічної стеганографії, що базуються на структурній особливості файлових систем у носіях інформації [11 – 14].

У роботі розглянуто третій напрямок, зокрема досліджуються методи приховування даних у структуру файлової системи сімейства FAT. Приховування реалізується шляхом перемішування кластерів певних покрівельних файлів (англ. Cover File). Пропонується удосконалений метод, який дозволяє значно збільшити обсяг прихованих даних за рахунок збільшення обчислювальної складності.

Приховування даних в кластерні файлові системи

Перші методи технічної стеганографії, які засновано на приховуванні інформаційних даних у структуру файлової системи, розглянуто в [11, 12]. Найпростіші методи застосовують вільні кластери (або певні службові поля даних) для запису прихованого повідомлення, але такий спосіб є ненадійним [13, 14]. Інші підходи застосовують надмірність, яка виникла штучно, у способі нумерації застосованих кластерів. Шляхом зміни нумерації окремих кластерів певних файлів (їх називають покрівельними файлами) вдається приховати невелику кількість інформаційних бітів [13, 14]. Розглянемо найбільш ефективний спосіб такого приховування з метою його подальшого розвитку та вдосконалення.

Для базового методу [13] застосуємо такі позначення. Нехай повідомлення M, яке буде вбудовано, позначається через масив $M=[b_0,b_1,...,b_{n-1}]$, де n – довжина повідомлення (кількість стеганоблоків), b_i – окремий стеганоблок (набір з m бітів), $i=0,1,...,n-1$. Покрівельні файли позначимо як $F_0,F_1,...,F_{p-1}$, де p – кількість покрівельних файлів, $p=2^m$ $m\in N$. Натуральне число m є ключовою інформацією.

Іменами (назвами) покрівельних файлів F_i є строки $t_i, i=0,1,...,\mathrm{p}-1$. Послідовність покрівельних файлів $F_i, i=0,1,...,\mathrm{p}-1$ (або їх назв $t_i, i=0,1,...,\mathrm{p}-1$) є ключовою інформацією.

Матриця $C=[\mathrm{c}_{ij}]$ містить номера кластерів покрівельних файлів. У такому випадку кожен покрівельний файл F_i може бути представлено у вигляді масиву

$$F_i=[\mathrm{c}_{i0},\mathrm{c}_{i1},...,\mathrm{c}_{iL_i}],$$

де L_i – загальна кількість кластерів файлу F_i.

У випадку, коли вбудовується псевдовипадкове повідомлення, довжина кожного покрівельного файлу F_i у кластерах повинна бути $L_i \approx n/m2^m$. У гіршому випадку, щоб гарантовано вбудувати повідомлення, довжина кожного покрівельного файлу F_i повинна бути $L_i \approx n/m$.

Для ***приховування повідомлення*** необхідно виконати наступну послідовність дій [13].

Повідомлення M розбивають на стеганоблоки по m біт кожен, $M=[b_0,b_1,...,b_{k-1}]$, де $k=n/m$. Якщо останній стеганоблок не повний, то необхідно доповнити його нульовими бітами.

Кожен блок b_j інтерпретується натуральним числом: $b_j \in N$, $j=0,1,...,k$; $0 \le b_j \le p-1$, яке відповідає певному покрівельному файлу $F_i, i=b_j$.

Виразити матрицю $C=[c_{ij}]$, у вигляді впорядкованого масиву $c_{ij}^0, c_{ij}^1, ...c_{ij}^w$, де $w=\sum_0^{p-1}(L_i)$. Тоді вбудовування повідомлення M задають у вигляді перестановки $\pi(c_{ij}^z)=c_{b_x y}$, де $z=0,1,...,w$; $x=0,1,...,n-1$; $y=0,1,...,L_y$. Величини x,y,z на початку роботи методу приймають нульові значення.

Повідомлення вважається вбудованим, якщо $z=w$ та $x \ge n-1$. Якщо $y>L_y$ та $x \le n-1$, то дане повідомлення не може бути вбудованим. Якщо $x>n-1$ та $z \le w$, то виконується перестановка з використанням кластерів файлів, які не брали участь у вбудовуванні повідомлення. Після кожної перестановки необхідно збільшити кожне значення $z=z+1$; $x=x+1$; $y=y+1$.

Для ***вилучення повідомлення*** необхідно мати ключову інформацію: m – натуральне число та набір покрівельних файлів $F_0, F_1, ..., F_{p-1}$. Далі необхідно виконати наступну послідовність дій [13].

Скомпоновати матрицю $C=[c_{ij}]$, що містить кластери покрівельних файлів. Дану матрицю необхідно виразити у вигляді впорядкованого масиву $c_{ij}^0, c_{ij}^1, ..., c_{ij}^w$ де $w=\sum_0^{p-1}(L_i)$.

Вилучене повідомлення $M^*=[b_0,b_1,...,b_w]$ містить стеганоблоки, вилучені із впорядкованого масиву C за таким правилом: $b_z=\pi^{-1}(c_{ij}^z)$, де $z=0,1,...,w$. Тобто значення блоку b_z дорівнює номеру покрівельного файлу кластера c_{ij}^z. Саме повідомлення вилучається шляхом конкатенації стеганоблоків $M=b_0|b_1|...|b_w$.

Для прикладу, якщо використовується два покрівельних файли ($p=2$), то значення кожного блоку b_i стеганограми може бути «0» або «1», де «0» відповідає кластеру, що належить до першого файлу, а «1» – до другого. Нехай покрівельні файли мають назви *A.txt* та *B.txt* і кожен з них займає по 10 кластерів у структурі файлової системи. Припустимо, що кластери цих файлів дефрагментовані, ланцюги кластерів файлів мають такі значення:

– для файлу *A.txt* маємо послідовність $\{3,4,5,6,7,8,9,10,11,12\}$;

– для файлу *B.txt* маємо послідовність $\{13,14,15,16,17,18,19,20,21,22\}$.

Нехай повідомлення, яке треба приховати, дорівнює 0x47. Розбивши це повідомлення на відповідні стеганоблоки, отримаємо двійковий масив $M=\{0,1,0,0,0,1,1,1\}$. Отже, виконавши етап перемішування кластерів у ланцюгах із відповідністю до масиву із стеганоблоків, отримаємо відповідні ланцюги кластерів:

– для файлу *A.txt* маємо нову послідовність $\{3,5,6,7,11,12,13,14,15,16\}$;

– для файлу *B.txt* також маємо нову послідовність $\{4,8,9,10,17,18,19,20,21,22\}$.

Отже приховане повідомлення міститься у зміненій нумерації окремих кластерів покрівельних файлів *A.txt* та *B.txt* , структуру файлової системи у спрощеному вигляді до та після приховування базовим методом наведено на рис. 1.

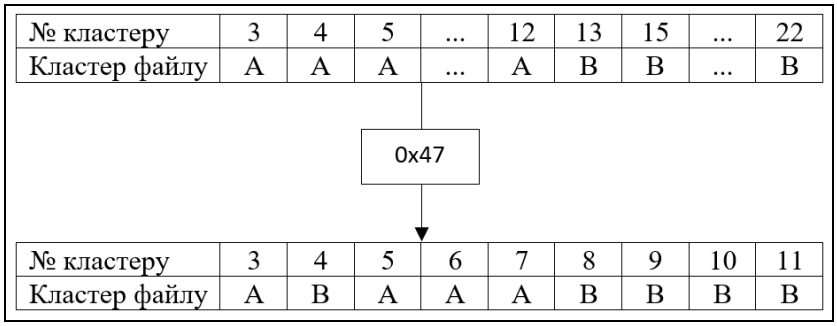

Рис. 1. Приклад приховування повідомлення 0x47 за рахунок перестановки кластерів

Слід відмітити переваги розглянутого методу. Перш за все це надійність приховування:

– обсяг інформації, що зберігається на носії, не змінюється;

– не змінюється кількість вільних чи помилкових кластерів;

– якщо носій був фрагментований, то після приховування рівень фрагментованості майже не змінюється [13, 14].

Отже, як стверджують розробники [13], немає ніяких зовнішніх ознак щодо детектування прихованого повідомлення. Збільшується лише рівень переплетеності файлів, але цей показник треба додатково досліджувати [13, 14].

До недоліків розглянутого методу слід віднести невеликий обсяг прихованих даних. Зокрема, при використанні $p = 2^m$ покрівельних файлів у кожному кластері можна приховати m інформаційних бітів. Далі пропонується удосконалений метод, застосування якого дозволяє збільшити цей показник.

Удосконалений метод приховування

Удосконалений метод приховування інформації використовує додаткову особливість розміщення файлів у структурі файлової системи, а саме – послідовність кластерів у межах одного покрівельного файлу. Тобто, зазвичай, кластери файлу $F_i = [c_{i0}, c_{i1}, ..., c_{iL_i}]$ впорядковані, так що $c_{iy} < c_{i(y+1)}$, $y = 0,1,..., L_i$. Нехай, якщо $c_{iy} < c_{i(y+1)}$ то це відповідає нульовому бітовому значенню – «0», якщо $c_{iy} > c_{i(y+1)}$, то це відповідає одиничному бітовому значенню – «1». Таким чином, перший кластер файлу не несе інформаційного значення, а є лише опорним кластером для наступних.

Для ***приховування повідомлення*** удосконаленим методом необхідно виконати наступну послідовність дій.

По-перше, необхідно повідомлення M розбити на стеганоблоки по m біт кожен, $M = [b_0, b_1, ..., b_{k-1}]$, де $k = n/m$. Причому кількість покрівельних файлів $p = 2^{m-1}$. Якщо $m \geq 2$, то повідомлення M необхідно умовно розбити на підповідомлення $M = Ml \,|\, Mr$, де $Ml = [bl_0, bl_1, ..., bl_{k-1}]$, $bl_i = b_i$ □ 1 – значення стеганоблоків без урахування найменш значущого біту, а $Mr = [br_0, br_1, ..., br_{k-1}]$, $br_i = b_i \,\&\, 0x1$ – значення найменш значущого біту кожного стеганоблоку, де $i = 0,1,..., k-1$. Після чого необхідно виконати перестановку впорядкованого масиву C за значеннями стеганоблоків повідомлення Ml як для базового методу.

Наступним кроком є перестановка кластерів у межах одного покрівельного файлу $F_i = [c_{i0}, c_{i1}, ..., c_{iL_i}]$. Першим етапом є встановлення опорного кластеру: $\pi(c_{i0}) = c_{i0}$, якщо $bl_j = i$ та $br_j = 0$, або $\pi(c_{i0}) = c_{iL_i}$, якщо $bl_j = i$ та $br_j = 1$ при $j = 0,1,...,k-1$.

Далі необхідно розглядати серії значень стеганоблоків br_j таких, що $\forall bl_j = i$ при $j = 0,1,...,k-1$ та $i = 0,1,..., p-1$. Якщо $br_j, br_{j+1}, ..., br_{j+k} = 0$, то виконується перестановка $\pi(c_{ji}) = c_{xi}$, $\pi(c_{(j+k)i}) = c_{yi}$, де $k \in N$ – розмір серії стеганоблоків із «0» значеннями, c_{xi} – перший кластер файлу F_i, який ще не брав участь у перестановці, c_{yi} – останній кластер файлу, який ще не брав участь у перестановці.

Якщо $br_j, br_{j+1}, ..., br_{j+k} = 1$, то виконується перестановка $\pi(c_{ji}) = c_{yi}$, $\pi(c_{(j+k)i}) = c_{xi}$, де $k \in N$ – розмір серії стеганоблоків із «0» значеннями, c_{xi} – перший кластер файлу F_i, який ще не брав участь у перестановці, c_{yi} – останній кластер файлу, який ще не брав участь у перестановці.

У разі, якщо усі інформаційні блоки br_i для файлу F_i були використані, то подальша перестановка виконується у довільному порядку.

Для ***вилучення повідомлення*** необхідно мати ключову інформацію: m – натуральне число та набір покрівельних файлів $F_0, F_1, ..., F_{p-1}$. Далі необхідно виконати наступну послідовність дій.

Необхідно скомпоновати матрицю $C = [c_{ij}]$, що містить кластери покрівельних файлів. Дану матрицю необхідно виразити у вигляді впорядкованого масиву $c_{ij}^0, c_{ij}^1, ..., c_{ij}^w$, де $w = \sum_0^{p-1}(L_i)$. Після чого вилучити масив стеганоблоків $M^* = [b_0, b_1, ..., b_w]$ як за базовим методом. Наступним кроком є конкатенація стегадоблоків із бітовим значенням, якщо $c_{ij}^x < c_{i(j+1)}^y$, то $b_z = b_z \,|\, 0$, та якщо $c_{ij}^x > c_{i(j+1)}^y$, то $b_z = b_z \,|\, 1$, при $x, y, z = 0,1,..., w$.

Для прикладу, якщо використовуються два покрівельних файли, то значення стеганоблоку може бути «00», «01», «10» або «11». Де «0…» відповідає кластеру, що належить до першого файлу, а «1…» – до другого. Нехай покрівельні файли мають назви *A.txt* та *B.txt*, кожен з них займає по 10 кластерів у структурі файлової системи, та ці кластери є дефрагментованими, тобто ланцюги кластерів файлів мають такі значення:

– для файлу *A.txt* маємо послідовність $\{3,4,5,6,7,8,9,10,11,12\}$;

– для файлу *B.txt* маємо послідовність $\{13,14,15,16,17,18,19,20,21,22\}$.

Нехай стеганограмою буде повідомлення 0x4747. Розбивши це повідомлення на відповідні стеганоблоки, отримаємо масив $M = \{01,00,01,11,01,00,01,11\}$.

Виконавши етап перестановки кластерів у ланцюгах із відповідністю до масиву із стеганоблоків, як для базового методу, отримаємо відповідні ланцюги кластерів:

– для файлу *A.txt* маємо нову послідовність $\{3,4,5,7,8,9,11,12,13,14\}$,

– для файлу *B.txt* також маємо нову послідовність $\{6,10,15,16,17,18,19,20,21,22\}$.

Виконавши наступний етап перестановки, отримаємо відповідні ланцюги кластерів:

– для файлу *A.txt* маємо $\{14,3,13,12,4,11,5,7,8,9\}$;

– для файлу *B.txt* маємо $\{22,21,6,10,15,16,17,18,19,20\}$.

Отже приховане повідомлення міститься у зміненій нумерації окремих кластерів покрівельних файлів *A.txt* та *B.txt*. В порівнянні із базовим методом змінено не лише відносну нумерацію кластерів окремих покрівельних файлів, але і відносну нумерацію окремих кластерів в кожному покрівельному файлі. Структуру файлової системи у спрощеному вигляді до та після приховування удосконаленим методом наведено на рис. 2.

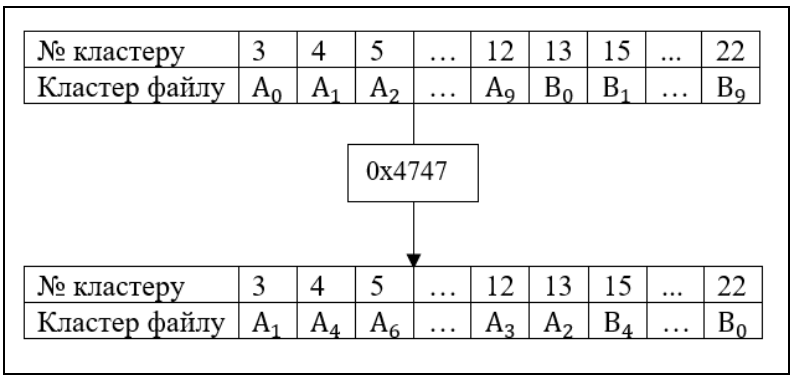

Рис. 2. Приклад приховування повідомлення 0x4747 удосконаленим методом

Порівняльний аналіз та оцінка ефективності удосконаленого методу

Для порівняльного аналізу розглянутих методів необхідно надати кількісну оцінку пропускної здатності організованого стеганоканалу. Під пропускною здатністю будемо мати на увазі максимально можливий розмір повідомлення, що приховується.

Пропускна здатність розглянутих методів залежить як від розміру одного кластеру, так і від кількості покрівельних файлів. Задля знаходження пропускної здатності припустимо, що:

– покрівельні файли займають усе вільне місце у структурі файлової системі;

– розміри покрівельних файлів рівні по відношенню один до одного;

– кількість кожного типу стеганоблоків однакова, наприклад, якщо стеганоблоки дорівнюють «0» та «1», то у повідомлені їх по 50 % кожного.

Приймемо такі позначення:

– $Data$ – загальний об’єм інформації (в байтах) у структурі файлової системі, що займають покрівельні файли, $Data = Const$;

– $Size_{Cl}$ – розмір одного кластеру у байтах, $Size_{Cl} \in \{2048, 4096, 8192, ..., 65536\}$;

– Num_{Cl} – загальна кількість кластерів, що займають покрівельні файли;

– $Size_{Fl}$ – розмір (у байтах) одного покрівельного файлу;

– Num_{Fl} – кількість покрівельних фалів;

– STG_{SIZE} – розмір стеганограми у байтах.

Так як $Data$ прийнято за константу, а покрівельні файли є рівнозначної величини, то можна стверджувати, що

$$Data = Size_{Fl} \times Num_{Fl}, \quad (1)$$

$$Data = Size_{Cl} \times Num_{Cl}. \quad (2)$$

Так як одним кластером можна відобразити один стеганоблок, а розмір одного стеганоблоку залежить (як логарифм двійковий) від кількості покрівельних файлів, то можна стверджувати, що розмір стеганограми у байтах

$$STG_{SIZE} = \frac{Num_{Cl} \times \log_2(Num_{Fl})}{8} \quad (3)$$

Для удосконаленого методу розмір одного стеганоблоку також залежить від кількості покрівельних файлів, але так як взято за увагу додаткову властивість, то кількість біт збільшується на 1 для кожного стеганоблоку, але так як перший кластер кожного покрівельного файлу не є кластером-повідомленням, то можна стверджувати, що розмір стеганограми у байтах

$$STG_{SIZE} = \frac{(Num_{Cl} - Num_{Fl}) \times \log_2(Num_{Fl} + 1)}{8} \quad (4)$$

Оцінимо *залежність пропускної здатності від розміру одного кластеру*, для цього зафіксуємо кількість покрівельних файлів, тобто $Num_{Fl} = Const$. А так як розмір покрівельних файлів однаковий, то й $Size_{Fl} = Const$. Із формул (1) та (2) можна вивести таку залежність:

$$Num_{Cl} = \frac{Size_{Fl} \times Num_{Fl}}{Size_{Cl}} \quad (5)$$

Надалі отриману залежність із формули (5) підставимо у формулу (3):

$$STG_{SIZE} = \frac{\frac{Size_{Fl} \times Num_{Fl}}{Size_{Cl}} \times \log_2(Num_{Fl})}{8} \quad (6)$$

Для спрощення аналізу у формулі (6) константні значення необхідно прийняти за одиницю та наближеними до нуля – знехтувати. $\log_2(Num_{Fl})$ – для коректності формули приймаємо рівним одиниці. Отримаємо, що $STG_{SIZE} = 1/Size_{Cl}$, тобто *чим більший розмір кластеру, тим менше повідомлення можна приховати, і як висновок – тим менша пропускна здатність методу*. Для удосконаленого методу залежність буде така сама.

Оцінимо *залежність пропускної здатності від кількості покрівельних файлів*, для цього зафіксуємо розмір кластеру, тобто $Size_{Cl} = Const$. А так як загальна кількість кластерів залежить від розміру кластеру, то й $Num_{Cl} = Const$. Узявши за основу формулу (3), замінимо константні значення одиницею та отримаємо, що $STG_{SIZE} = \log_2(Num_{Fl})$. Тобто, *чим більше покрівельних файлів, тим більше повідомлення можна приховати, і як висновок – тим більша пропускна здатність методу*. Для удосконаленого методу залежність така сама але розмір стеганограми у два рази більший при відповідних умовах. Тобто при рівних умовах, *використовуючи удосконалений метод, можна приховати у два рази більше інформації* (якщо знехтувати константними величинами). Можна стверджувати, що дана рівність $STG_{SIZE} = \log_2(Num_{Fl} + 1)$ вірна для удосконаленого методу.

Виявивши залежності, для більш коректної оцінки підставимо числові значення замість $Data$, $Size_{Cl}$, Num_{Fl}. Припустимо, що розглянуті методи будуть застосовані для приховування інформації у змінному флеш-накопичувачі. Нехай загальний розмір флеш-накопичувача займає 8 Гб, тобто $Data = 7780302848$ байт, кількість покрівельних файлів $Num_{Fl} = 2$, та розмір одного кластеру $Size_{Cl} = 2048$ байт. Тоді загальна кількість кластерів $Num_{Cl} = \frac{7780302848}{2048} = 3799952$. Підставивши отримані значення у формулу (3), знайдемо розмір стеганограми у байтах: $STG_{SIZE} = \frac{3799952}{8} = 474994$ байт.

Зафіксувавши кількість файлів, знайдемо розмір стеганограми відповідно для розміру кластеру $Size_{Cl} \in \{2048, 4096, 8192, ..., 65536\}$, отримані результати зведено до табл. 1.

Таблиця 1

Залежність розміру стеганограми від розміру кластеру

Розмір кластеру, байт	Кількість кластерів	Базовий метод, байт	Удосконалений метод, байт
2048	3798976	474872	949744
4096	1899488	237436	474872
8192	949744	118718	237436
16384	474872	59359	118718
32768	237436	29680	59359
65536	118718	14840	29680

Побудувавши графік з використанням даних табл. 1, можна впевнитись що, дійсно, розмір стеганограми залежить від розміру одного кластеру у зворотній пропорційності, як це зображено на рис. 3. Крім того, з рис. 3 наочно видно переваги удосконаленого методу, а саме – вдвічі збільшену пропускну спроможність організованого стеганоканалу.

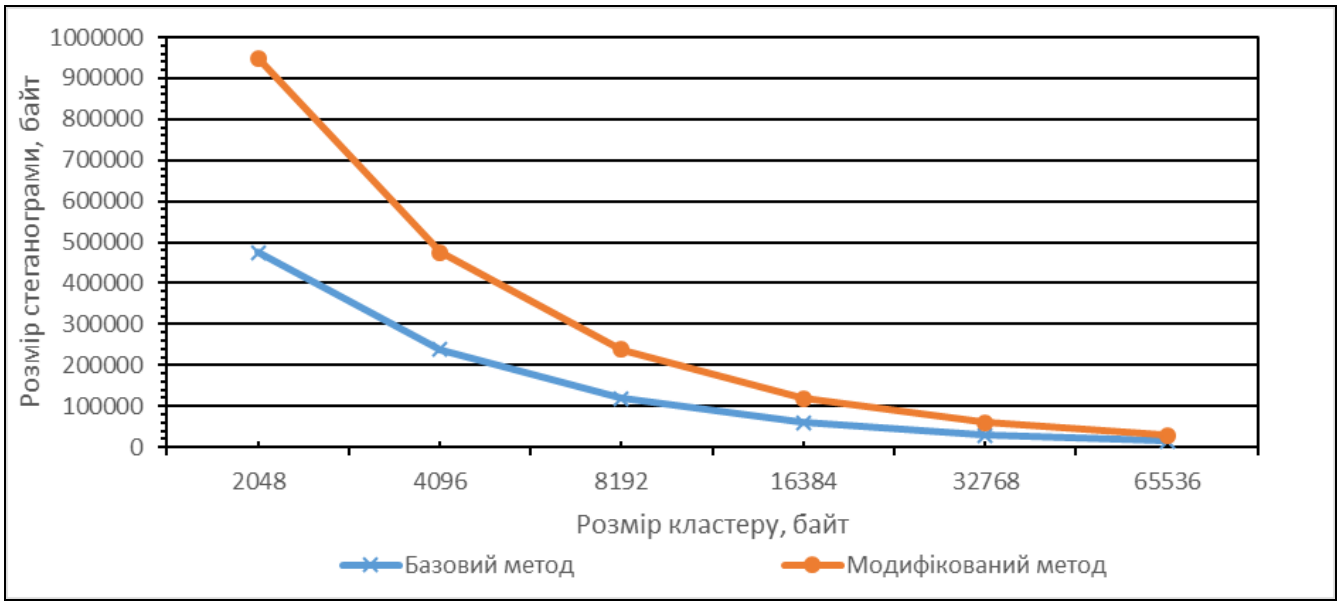

Рис. 3. Графік залежності розміру стеганограми від розміру кластеру

Аналогічну залежність виявимо і від кількості покрівельних файлів, для цього зафіксуємо загальний розмір накопичувача $Data = 7780302848$, та розмір одного кластеру – $Size_{Cl} = 2048$, кількість покрівельних файлів прийматиме такі значення – $Num_{Fl} \in \{2,4,8,16,32,64\}$. Результати розрахунків зведено до табл. 2.

Таблиця 2

Залежність розміру стеганограми від кількості покрівельних файлів

Кількість покрівельних файлів	Розмір одного файлу, байт	Базовий метод, байт	Удосконалений метод, байт
2	3 890 151 424	474872	949744
4	1945075712	949744	1424616
8	972537856	1424616	1899488
16	486268928	1899488	2374360
32	243134464	2374360	2849232
64	121567232	2849232	5698464

Згідно з даними табл. 2 побудуємо графік залежності розміру стеганограми від кількості покрівельних файлів, результати наведено на рис. 4.

Порівнюючи результати з табл. 1 та 2, можна стверджувати, що при рівних вхідних параметрах удосконалений метод дозволяє приховувати повідомлення вдвічі більшого розміру ніж базовий метод. До того ж удосконалений метод дозволяє використовувати лише один покрівельний файл, на відміну від базового методу.

Для оцінки обчислювальної складності необхідно визначити базові операції розглянутих методів.

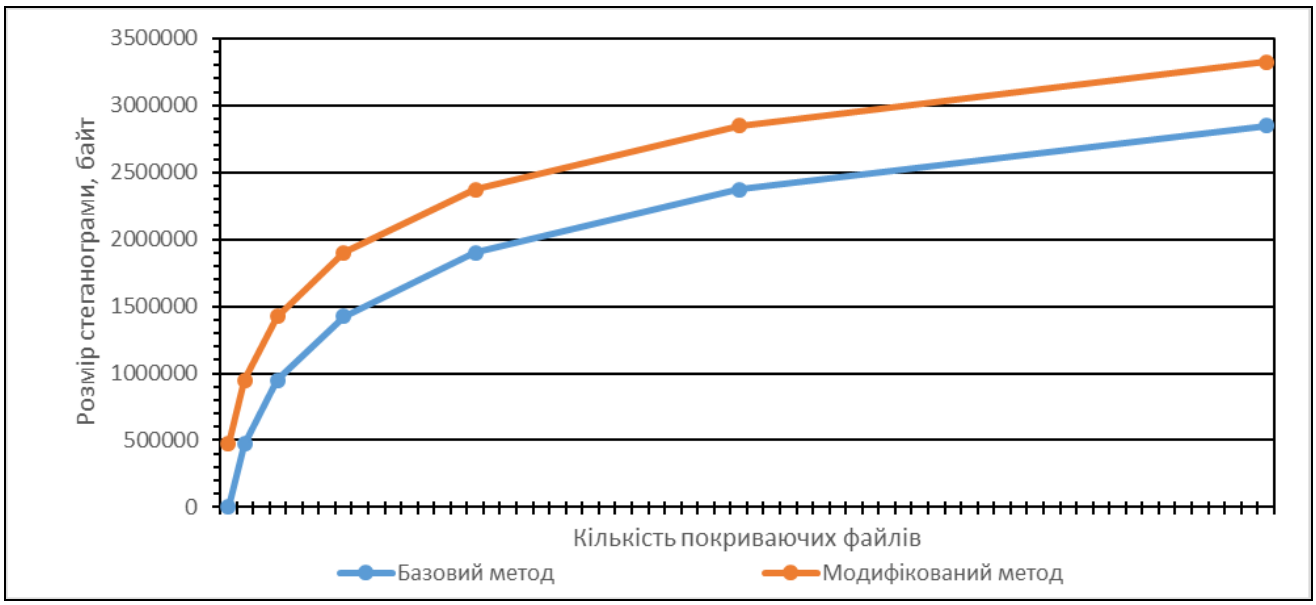

Рис. 4. Графік залежності розміру стеганограми від кількості покрівельних файлів

Оцінюючи опис методів, можна стверджувати, що базовою операцією є перестановка кластерів – $\pi(c)$. Таким чином обчислювальна складність безпосередньо залежить від кількості кластерів, які необхідно переставити. У найгіршому випадку необхідно виконати перестановку усіх кластерів покрівельних файлів, тобто $\pi_0^w(c_{ij})$, де w – сумарна кількість кластерів покрівельних файлів.

Для базового методу необхідно виконати перестановку один раз, тобто обчислювальна складність знаходиться у лінійній залежності від кількості кластерів, які необхідно переставити – $O(n)$.

Для удосконаленого методу необхідно спочатку виконати перестановку за базовим методом, а потім виконати перестановку в межах кожного покрівельного файлу. Тобто, у найгіршому випадку, кожен кластер може бути переставлено двічі. Таким чином обчислювальна складність знаходиться у лінійній залежності від кількості кластерів, які необхідно переставити – $O(2n)$.

Порівнюючи обчислювальні складності методів, можна стверджувати, що для удосконаленого методу необхідно у двічі більше обчислювальних ресурсів.

Для експериментальних досліджень ефективності розглянутих методів була використана програма «SteganoFAT», файлова система FAT32 на флеш накопичувачу JetFlash 350 Transcend ємністю 8 Гб, інтерфейс підключення USB2.0 та ноутбук Lenova Y510P із операційною системою Windows 8.1. Необхідно зазначити, що фактичний час реалізації методів приховування залежить як від апаратних особливостей носіїв інформації, так й від алгоритмічної реалізації. Проаналізуємо час роботи методів, у залежності від обраних параметрів:

– розмір кластеру;
– розмір файлу повідомлення;
– кількість покрівельних файлів;
– загальний розмір покрівельних файлів;

Для аналізу залежності затраченого часу на виконання методів приховування та вилучення повідомлення, від розміру кластеру зафіксуємо розмір повідомлення – 100 байт, кількість покрівельних файлів – 2, загальний розмір покрівельних файлів – 7 Мб. Будемо змінювати розмір кластеру – 2048, 4096, 8192 байт. Отримані результати експериментальних досліджень зведено у табл. 3. В цій таблиці і далі наводиться час, витрачений базовим методом та, через дріб, час, витрачений удосконаленим методом.

Таблиця 3

Часові витрати в залежності від розміру кластеру

Розмір одного кластеру, байт	2048	4096	8192
Час приховування повідомлення, с	3.341/6.276	2.87/5.41	2.37/3.96
Час вилучення повідомлення, с	0.022/0.021	0.012/0.012	0.008/0.009

Як видно із табл. 3, при збільшенні розміру кластеру час на приховування повідомлення зменшується. Удосконалений метод потребує вдвічі більше часу на приховування інформації, ніж базовий, при тій самій конфігурації. На час вилучення інформації обраний метод не впливає.

Для оцінки залежності витраченого часу на виконання приховування та вилучення повідомлення від розміру повідомлення зафіксуємо розмір кластеру – 2048 байт, кількість покрівельних файлів – 2, загальний розмір покрівельних файлів 7 Мб. Будемо змінювати розмір повідомлення: 100, 200, 400 байт. Результати часового аналізу вказані у табл. 4.

Таблиця 4

Часові витрати в залежності від розміру повідомлення

Розмір повідомлення, байт	100	200	400
Час приховування повідомлення, с	3.82/7.34	5.84/10.12	8.36/15.57
Час вилучення повідомлення, с	0.02/0.02	0.02/0.02	0.03/0.03

Як видно із табл. 4, при збільшенні розміру повідомлення час на приховування та вилучення повідомлення збільшується.

Для оцінки залежності витраченого часу на виконання приховування та вилучення повідомлення від кількості покрівельних файлів зафіксуємо розмір кластеру – 2048 байт, розмір повідомлення – 100 байт, загальний розмір покрівельних файлів – 7 Мб. Будемо змінювати кількість покрівельних файлів: 2, 4, 8. Отримані результати зведено у табл. 5.

Таблиця 5

Часові витрати в залежності від кількості покрівельних файлів

Кількість покрівельних фалів	2	4	8
Час приховування повідомлення, с	4.693/8.601	2.76/5.31	2.704/5.01
Час вилучення повідомлення, с	0.022/0.017	0.025/0.026	0.032/0.031

Як видно із табл. 5, при збільшенні кількості покрівельних файлів час на приховування повідомлення зменшується. Це пов’язано із тим, що кількість інформаційних кластерів при збільшенні кількості покрівельних файлів зменшується та відповідно збільшується кількість кластерів, які будуть записані впорядковано, а не перемішано. При вилученні повідомлення затрачений час збільшується із кількістю покрівельних файлів.

Для оцінки останньої залежності витраченого часу на виконання приховування та вилучення повідомлення від загального розміру покрівельних файлів зафіксуємо розмір кластеру – 2048 байт, кількість покрівельних файлів – 2, розмір повідомлення 100 байт. Будемо змінювати загальний розмір покрівельних файлів: 1.7, 3.5, 7 Мб. Отримані результати наведено у табл. 6.

Як видно із табл. 6, при збільшенні загального розміру покрівельних файлів час на приховування та вилучення повідомлення збільшується.

Таблиця 6

Часові витрати від загального розміру покрівельних файлів

Загальний розмір покрівельних файлів, Мб	1.7	3.5	7
Час приховування повідомлення, с	2.083/4.201	2.417/5.01	3.293/6.71
Час вилучення повідомлення, с	0.007/0.006	0.012/0.013	0.021/0.02

Висновки

Запропоновано та досліджено удосконалений метод приховування та вилучення інформаційних даних у структуру файлової системи сімейства FAT. Роблячи висновки із отриманих експериментальних результатів, можна стверджувати:

- на час приховування та вилучення повідомлення здебільшого впливає кількість кластерів, над якими необхідно виконати перестановку;
- вилучення виконується значно швидше за приховування повідомлення, це пов'язано із відсутністю необхідності виконувати перестановку над кластерами під час вилучення повідомлення;
- час вилучення повідомлення не залежить від обраного методу.

Таким чином, доцільним є використання обох методів при приховуванні інформації у структуру файлової системи. Вибір методу приховування залежить від: наявності часу для приховування та вилучення повідомлення; наявності обчислювальних ресурсів; вхідних параметрів методу (розмір та структура повідомлення, кількість та розміри покрівельних файлів). Ці та інші налаштування потребують подальших розробок та дослідження.

Список літератури:

1. Горбенко І.Д., Горбенко Ю.І. Прикладна криптологія. Теорія. Практика. Застосування : підручник для вищих навч. закладів. Харків : Форт, 2013. 880 с.
2. Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone. Handbook of Applied Cryptography – CRC Press, 1997. 794 p.
3. N. Ferguson and B. Schneier. Practical Cryptography. John Wiley & Sons, 2003, 432 p.
4. Petitcolas F. Information Hiding / F. Petitcolas, R. J.Anderson, M. G. Kuhn // Proceedings IEEE. 1999. Vol. 87, №. 7. pp. 1069-1078.
5. Конахович Г.Ф., Пузиренко О.Ю. Компьютерная стеганография. Теория и практика. К. : МК-Пресс, 2006. 288 с.
6. Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. Москва : Солон-Пресс, 2002. 272 с.
7. Хорошко В.А., Шелест М.Е. Введение в компьютерную стеганографию. К., 2002. 140 с.
8. Горбенко І.Д. Захист інформації в ІТС. Криптографічний захист інформації : навч. посібник. Харків : ХНУРЕ, 2004. 376 с.
9. Пескова О.Ю., Халабурда Г.Ю. Применение сетевой стеганографии для защиты данных, передаваемых по открытым каналам интернет // Информ. системы для науч.х исследований (IMS-2012). Тр. XV Всерос. объединенной конф. "Интернет и современное общество" (IMS-2012). Санкт-Петербургский нац. иссл. ун-т информ. технологий, механики и оптики, 2012. С. 348-354. http://ojs.ifmo.ru/index.php/IMS/article/download/132/132
10. Кузнецов А.А., Коваленко О.Ю. Стеганографическая защита информации с использованием 3D-печати // Інформаційна безпека держави, суспільства та особистості : зб. тез доповідей Всеукр. наук.-практ. конф., 16 квітня 2015 року. Кіровоград : КНТУ, 2015. С. 91-92.
11. Hassan Khan, Mobin Javed, Syed Ali Khayam, Fauzan Mirza. Designing a cluster-based covert channel to evade disk investigation and forensics. Computers & Security Volume 30, Issue 1, January 2011.
12. Hassan Khan, Mobin Javed, Fauzan Mirza. Evading Disk Investigation and Forensics using a Cluster-Based Covert Channel. National University of Science & Technology (NUST), Islamabad 44000, Pakistan.
13. Nerijus Morkevičius, Grigas Petraitis, Algimantas Venčkauskas, Jonas Čeponis. Covert Channel for Cluster-based File Systems Using Multiple Cover Files. Information Technology and Control, 2013, Vol.42, No.3. – p 32.
14. Кузнецов А.А., Швагер А.С., Фесенко Д.А. Сокрытие данных в кластерных файловых системах // Радиотехника. 2015. Вып. 181. С. 86-100.

Харківський національний
університет імені. В.Н. Каразіна

Надійшла до редколегії 15.03.2018