УДК 004.056.5 **DOI:10.30837/rt.2021.3.206.07**

К.Ю. ШЕХАНІН, С.В. ПШЕНИЧНА, О.О. КУЗНЕЦОВ, д-р техн. наук

ДОСЛІДЖЕННЯ ОБЧИСЛЮВАЛЬНОЇ СКЛАДНОСТІ МЕТОДІВ ПРИХОВУВАННЯ ІНФОРМАЦІЇ У КЛАСТЕРНІ СТЕГАНОСИСТЕМИ

Вступ

На сьогодні інформацію розглядають як один з основних ресурсів для розвитку сучасного суспільства, а інформаційні системи та технології – як засоби підвищення ефективності та продуктивності роботи сучасних систем.

Інформаційні технології визначають процеси передачі і розповсюдження, зберігання та обробки інформації, а також її використання у певних цілях. Інколи факт виконання цих процесів має бути прихований від сторонніх осіб. Цим і займається галузь науки стеганографія.

Суспільству здавна відома більшість стеганографічних методів, заснованих на фізичних явищах природи чи фізіологічних особливостей людського організму. Але технології не стоять на місці, із відкриттям нових засобів обробки та зберігання інформації з'являються нові методи приховування інформації, що засновані на технічних особливостях технологічних засобів і методів обробки інформації, дана галузь науки називається технічною стеганографією.

На даний час відомо декілька методів технічної стеганографії. Наприклад, приховування інформації у модель під час 3D-друку [1 – 3]. Дана галузь приховування інформації має певні переваги та недоліки, а саме: відносно більшу вартість при створенні прихованого повідомлення та складності при зчитуванні інформації. Другий напрямок технічної стеганографії пов'язаний із мережевим трафіком [4 – 7]. У даному методі інформація може приховуватись, наприклад, у поля заголовків протоколів, чи, наприклад, приховане повідомлення шляхом передається через посилання певної послідовності пакетів. Також існують методи приховування інформації у структуру файлової системи [8 – 10], але відомі методи, які або здатні приховати малу кількість інформації, або мають належний рівень стійкості до детектування. Таким чином, актуальною задачею є розробка методу приховування інформації, який здатний приховати більшу кількість інформації та має більший рівень стійкості до детектування, із задовільним рівнем обчислювальної складності.

У роботах [11, 12] представлено методи технічної стеганографії, що базуються на структурній особливості файлових систем у носіях інформації. А саме, приховування інформації у файловій системі FAT шляхом перемішування кластерів певних, ключових файлів. Методи приховування інформації у структуру кластерної файлової системи шляхом перемішування кластерів покриваючих файлів потребують значних обчислювальних ресурсів.

У даній роботі досліджено методи підвищення обчислювальної ефективності за кількістю необхідної оперативної пам'яті та за кількістю необхідних операцій для приховування повідомлення.

Обчислювальна складність методів приховування інформації

Необхідно визначити обчислювальну складність методів приховування та вилучення інформації шляхом перестановки кластерів покриваючих файлів структури файлової системи FAT для подальших рекомендацій щодо використання методів та розробки програмної реалізації.

Так як більшість часу на приховування повідомлення у структуру файлової системи займає саме робота із фізичним носієм, то виділимо такі операції:

– обчислювальна складність на переміщення зчитуючої головки фізичного носія (для HDD накопичувачів) чи зміна позиції робочого сектору (для SSD накопичувачів) – $O\left(f\left(n\right)\right)$,

де n – кількість переміщень зчитуючої головки фізичного носія у кількості пройдених секторів;

– обчислювальна складність на зчитування даних із сектору – $O\left(g\left(n\right)\right)$, де n – кількість кластерів що зчитані;

– обчислювальна складність на запис даних у сектор – $O\left(h\left(n\right)\right)$, де n – кількість кластерів що записано;

У деяких випадках обчислювальна складність залежить від кількості стеганоблоків – k. Далі необхідно визначити загальну обчислювальну складність для базового [11] та для модифікованого методів [11 – 13]. Обчислювальну складність f, g, h вважатимемо складністю за часовими ознаками, у той час як необхідну кількість оперативної пам'яті m – вважатимемо складністю за об'ємними параметрами.

Обчислювальною складністю на генерацію перестановки будемо нехтувати, так як час на виконання даної операції, у порівняні на час роботи із фізичним носієм, є мінімальним.

Загальна обчислювальна складність для базового методу складається із суми перелічених вище елементів обчислювальної складності:

$$O\left(B\right)=O\left(f\left(n\right)\right)+O\left(g\left(n\right)\right)+O\left(h\left(n\right)\right) \tag{1}$$

Також необхідно зазначити, що загальна обчислювальна складність залежить від кількості секторів що були перезаписані, та від кількості переміщень зчитуючої головки пристрою, а отже необхідно виявити залежність між розміром повідомлення (кількістю стеганоблоків) та кількістю перезаписаних кластерів. Для цього спиратимемося на роботу [11 – 14] приховування даних у структуру файлової системи.

Приховування інформації стеганографічними методами [11, 12] виконується за рахунок перезапису даних із кластерів. Та для того щоб покриваючи файли були цілісними інформацію, необхідно копіювати повністю, це робиться із використанням оперативної пам'яті. За способом роботи із оперативною пам'яттю можна виділити такі варіації:

– повний запис даних з кластерів до оперативної пам'яті, у такому випадку необхідна кількість оперативної пам'яті залежить від кількості кластерів покриваючих файлів;

– почерговий запис даних з кластерів до оперативної пам'яті, у такому випадку у оперативну пам'ять зчитуються дані з кластеру *Б*, після чого записуються дані з кластеру *А*, далі каретка зчитуючої головки переміщується до кластеру *В* і процес повторюється (зчитуються дані із *В*, записуються дані із *Б* і так далі). Таким чином, при виконанні приховування інформації необхідно мати розмір оперативної пам'яті як подвійний розмір до одного кластеру.

За шляхом генерації перестановок, можна виділити такі варіації виконання методів приховування:

– виконання перестановки із подальшим послідовним перезаписом кластерів у необхідній послідовності. Це означає, що спочатку приховуються кластери, які складають стеганограму, а вже після цього необхідно розмістити усі вільні (ті, які не несуть інформаційного навантаження на приховування повідомлення) кластери покриваючих файлів у впорядкованій послідовності. Спочатку впорядковані кластери першого покриваючого файлу, потім другого і так далі. Така варіація методу приховування дозволяє знизити надлишковий рівень фрагментації покриваючих файлів;

– виконання перестановки із переміщенням лише інформативних кластерів у нормальну послідовність. У такому випадку спочатку переміщуємо інформаційні кластери у відповідну до повідомлення послідовність, після чого переміщуємо кластери, які були витиснені інформаційними кластерами. Витиснені кластери впорядковано розміщуємо лише на позиції, де знаходилися інформаційні кластери. Перевагою такого методу є те, що необхідно буде перемістити лише обмежену розміром повідомлення кількість кластерів;

– виконання перестановки із оптимальним переміщенням лише інформативних кластерів. У такому випадку переміщуємо інформаційні кластери у відповідну до повідомлення послідовність, але із можливістю збереження позиції кластером, якщо він відповідає значенню стеганоблока. Усе інше аналогічно до попереднього методу. Такий метод дозволяє ще значніше зменшити кількість перезаписів кластерів, але можуть виникати випадку, коли покриваючі файли матимуть переплетеність (деякі кластери файлу розміщені у зворотній послідовності, що є аномальною поведінкою файлової системи і може детектувати приховане повідомлення).

Комбінуючи методи оптимізації за оперативною пам'яттю та методи зменшення обчислювальної складності, ми запропонували чотири способи приховування інформації у структуру кластерних файлових систем (чотири для базового методу та чотири – для модифікованого) із прикладами.

ПЗОП

Виконання перестановки із повним завантаженням кластерів до оперативної пам'яті та подальшим послідовним перезаписом кластерів у необхідній послідовності, рис. 1. Далі даний метод називатимемо повним завантаженням до оперативної пам'яті ПЗОП.

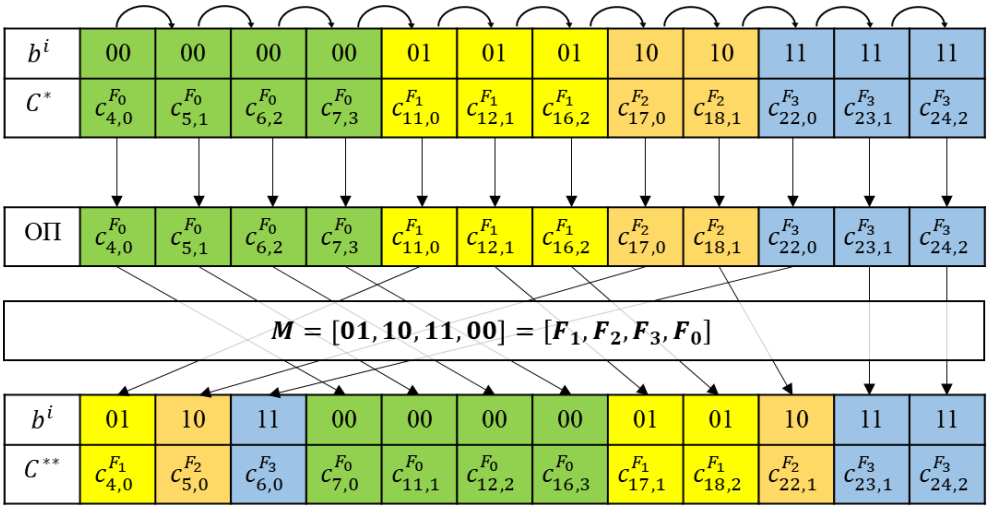

Рис.1. Приклад перестановки кластерів із повним завантаженням даних у операційну пам'ять

Даний метод реалізацій перестановки дозволяє нівелювати обчислювальну складність на переміщення зчитуючої головки фізичного носія, так як спочатку виконується послідовне зчитування кластерів до операційної пам'яті. Потім перестановка виконується у операційній пам'яті як робота із масивом, а далі так само послідовно виконується перезапис кластерів. Необхідно зазначити, що спосіб перемішування кластерів потребує значного розміру оперативної пам'яті, щоб завантажити усі кластери усіх покриваючих файлів до неї. А отже $O\left(m\left(n\right)\right)=n\times Cluster_{size}$, що означає, що може виникнути ситуація, коли операція перестановки не може бути виконана взагалі. Але як висновок необхідно буде перезаписати усі кластери, а отже загальна обчислювальна складність матиме вигляд

$$O\left(B\right)=O\left(f\left(2n\right)\right)+O\left(g\left(n\right)\right)+O\left(h\left(n\right)\right);n=C_{len}, \quad (2)$$

де C_{len} – довжина матриці стану у кластерах, тобто загальний розмір покриваючих файлів у кластерах. Подвійне переміщення зчитуючої головки як раз і пов'язане із тим, що необхідно спочатку зчитати усі кластери, а потім записати усі кластери.

Для даного прикладу матимемо такі показники, що зазначені у табл. 1, де $\varphi(F_i)$ означає кількість фрагментів та рівень фрагментації (відстань між фрагментами), відповідно, а π – правило перестановки:

Таблиця 1

Показники результату приховування прикладу шляхом ПЗОП

π	(4,7,16,18,22,6,12,17,5,11)(23)(24)
$O(f(n))$	20
$O(g(n))$	10
$O(h(n))$	10
$O(m(n))$	10
$\varphi(F_0)$	1
$\varphi(F_1)$	2 (6)
$\varphi(F_2)$	2 (7)
$\varphi(F_3)$	2 (7)

ПчЗОП-І

Виконання перестановки із почерговим завантаженням кластерів до оперативної пам'яті із впорядкованим розміщенням залишку кластерів, рис. 2. Далі даний метод називатимемо почерговим завантаженням до оперативної пам'яті ПчЗОП-І.

Особливість даного методу полягає у тому, що перестановка виконується почергово по ланцюгу кластерів, тобто на першій ітерації зчитується перший кластер ланцюгу, у наступній ітерації зчитується кластер, на який слідом перезаписуємо попередньо зчитаний кластер. Таким чином зростає кількість переміщень зчитуючої головки пристрою, так як доводиться "стрибати" назад та вперед по кластерах. Це означає, що кількість переміщень може бути максимум сумою чисел до числа n, де n – кількість кластерів, а отже $O(f(n^2))$, тобто складність має квадратичний характер. І так як усе одно виконується впорядкований перезапис подальших кластерів, то обчислювальна складність буде дорівнювати

$$O(B) = O(f(n^2)), +O(g(n)) + O(h(n)); n = C_{len} \quad (3)$$

де C_{len} – довжина матриці стану у кластерах, тобто загальний розмір покриваючих файлів у кластерах. Причому можуть виникати умови, коли кластер перезаписується сам у себе; у такому випадку виконувати перезапис кластеру не є необхідним. Також необхідно зазначити, що при виконанні перестановки методом ПчЗОП-І обчислювальна система потребує лише подвійного розміру кластеру у якості оперативної пам'яті, тобто $O(m(const)) = 2$, що дозволяє виконувати приховування інформації теоретично у необмежені за розміром покриваючі файли. Для даного прикладу матимемо такі показники, що зазначені у табл. 2.

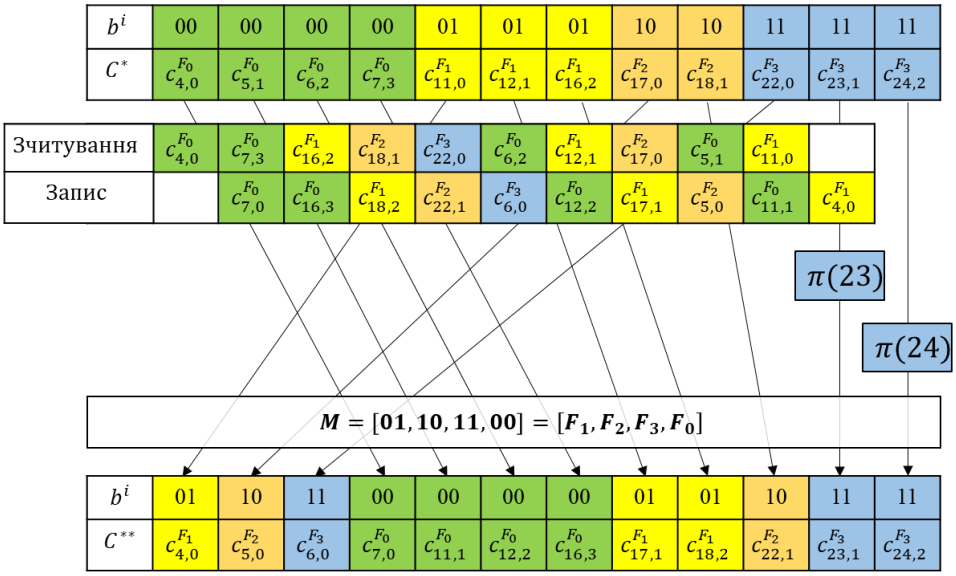

Рис. 2. Приклад перестановки кластерів із почерговим завантаженням кластерів ПчЗОП-І

Таблиця 2

Показники результату приховування прикладу шляхом ПчЗОП-І

π	(4,7,16,18,22,6,12,17,5,11)(23)(24)
$O\left(f\left(n^2\right)\right)$	34
$O\left(g\left(n\right)\right)$	10
$O\left(h\left(n\right)\right)$	10
$O\left(m\left(const\right)\right)$	2
$\varphi\left(F_0\right)$	1
$\varphi\left(F_1\right)$	2 (6)
$\varphi\left(F_2\right)$	2 (7)
$\varphi\left(F_3\right)$	2 (7)

ПчЗОП-ІІ

Виконання перестановки із почерговим завантаженням кластерів до оперативної пам'яті із переміщенням лише інформативних кластерів у нормальну послідовність, рис. 3. Далі даний метод називатимемо почерговим завантаженням до оперативної пам'яті ПчЗОП-ІІ.

Особливість даного способу приховування інформацій полягає у тому, що при розрахунку остаточної матриці стану кластерів перемішування виконується лише над кластерами, які безпосередньо беруть участь у приховуванні інформаційного повідомленні. Можуть переміщуватися неінформаційні кластері лише у випадку, коли їх заміщують інформаційні кластери. Наступним кроком є розміщення переміщених кластерів у нормальній послідовності, тобто кластери одного файлу повинні мати індекси, що зростають зліва направо, тобто не мати переплетеності між кластерами. Це дозволяє зменшити рівень фрагментації для можливо переплетених покриваючих файлів.

У даному випадку кількість переміщень зчитуючої головки та кількість зчитувань та записів кластерів вже залежить від кількості стеганоблоків – k, а не лише від кількості кластерів покриваючих файлів – n. У той час оперативній пам'яті необхідно так само лише подвійний розмір кластеру.

Для того щоб виконати перестановку, необхідно перемістити лише k кластерів, які можуть замістити собою ще k кластерів. Дані кластери можуть розміщуватися по усій довжині

матриці стану, тобто переміщення зчитуючої головки для переміщення одного кластеру може бути із крайньої лівої позиції до крайньої правої, тобто необхідно пройтись по усіх кластерах – n. А отже кількість переміщень зчитуючої головки дорівнює – $O\left(f\left(2kn\right)\right)$. Причому, якщо розмір повідомлення у стеганоблоках наближається до кількості кластерів, то обчислювальна складність переміщень зчитуючої головки наближається до квадратичної залежності – $O\left(f\left(n^2\right)\right); k \to n$.

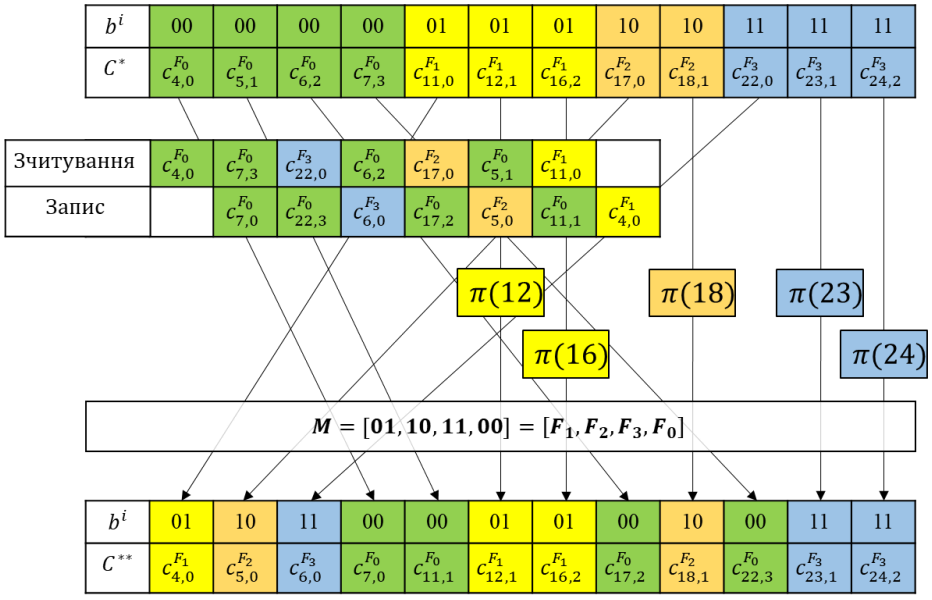

Рис. 3. Приклад перестановки кластерів із почерговим завантаженням кластерів ПчЗОП-II

Кількість зчитувань та записів кластерів фіксована та залежить лише від кількості стеганоблоків, але таких записів може бути по два, коли інформаційний кластер заміщує не інформаційній. А отже $O\left(g\left(2k\right)\right)$ та $O\left(h\left(2k\right)\right)$. Результуюча обчислювальна складність

$$O(B) = O\left(f\left(2kn\right)\right) + O\left(g\left(2k\right)\right) + O\left(h\left(2k\right)\right); n = C_{len}; k = M_{len}. \quad (4)$$

Так як не виконується подальше перерозміщення неінформативних кластерів у суцільні послідовності, то зазвичай рівень фрагментації покриваючих файлів при перерозміщенні у такий спосіб буде вищий, аніж при виконанні методу приховування інформації шляхом ПчЗОП-I. Для даного прикладу матимемо показники, що зазначені у табл. 3.

Таблиця 3

Показники результату приховування прикладу шляхом ПчЗОП-II

π	(4,7,22,6,17,5,11)(12)(16)(18)(23)(24)
$O\left(f\left(n^2\right)\right)$	34
$O\left(g\left(2k\right)\right)$	8
$O\left(h\left(2k\right)\right)$	8
$O\left(m\left(const\right)\right)$	2
$\varphi\left(F_0\right)$	3 (3)
$\varphi\left(F_1\right)$	2 (4)
$\varphi\left(F_2\right)$	2 (6)
$\varphi\left(F_3\right)$	2 (7)

ПчЗОП-III

Виконання перестановки із почерговим завантаженням кластерів до оперативної пам'яті із оптимальним переміщенням лише інформативних кластерів, рис. 4. Далі даний метод називатимемо почерговим завантаженням до оперативної пам'яті ПчЗОП-III.

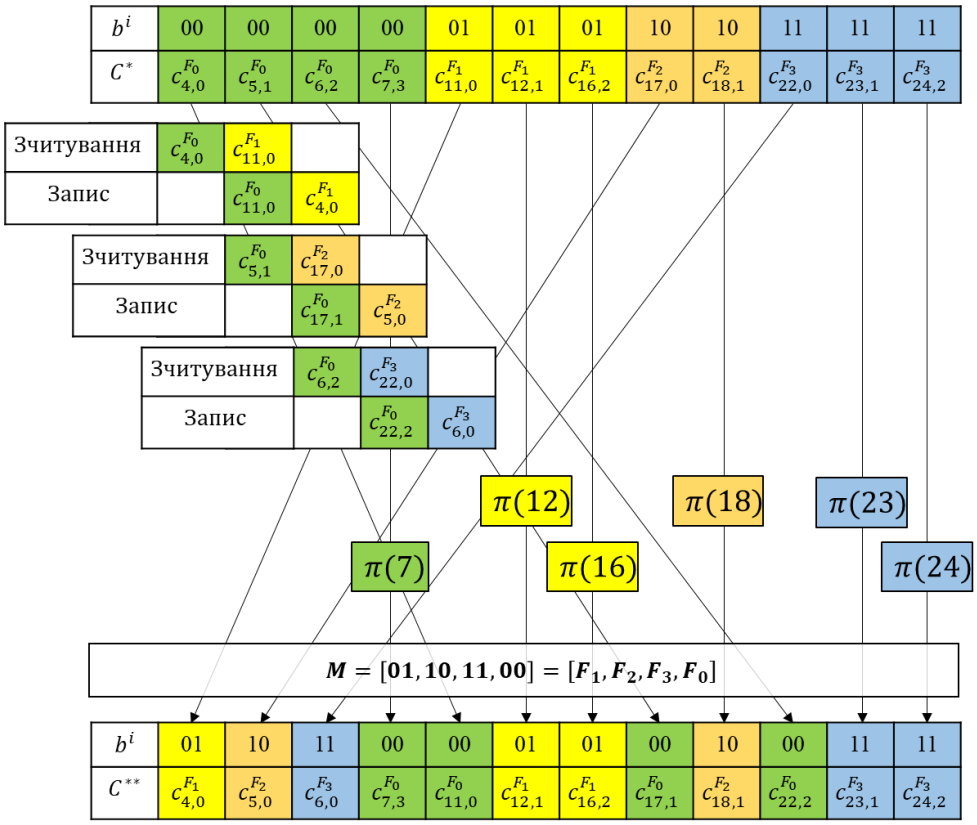

Рис. 4. Приклад перестановки кластерів із почерговим завантаженням кластерів ПчЗОП-III

Особливість даного способу приховування інформацій полягає у тому, що при розрахунку остаточної матриці стану кластерів перемішування виконується лише над кластерами, які безпосередньо беруть участь у приховуванні інформаційного повідомленні, але кластер файлу у початковому стані співпадає із кластером файлу у кінцевому стані, то такий кластер не переміщується. Така особливість може призвести до зростання переплетеності покриваючих файлів, що збільшить рівень фрагментації. Але така особливість надає перевагу, так як зменшиться кількість переміщень та перезаписів у порівнянні із способом ПчЗОП-II. Можуть переміщуватися неінформаційні кластері лише у випадку, коли їх заміщають інформаційні кластери. Також така особливість (ПчЗОП-III) має сенс, якщо використовується модифікований метод приховування інформації, так як він за визначенням призведе до зростання переплетеності.

У даному випадку кількість переміщень зчитуючої головки та кількість зчитувань та записів кластерів вже залежить від кількості стеганоблоків – k, а не лише від кількості кластерів покриваючих файлів – n. У той час оперативній пам'яті необхідно так само лише подвійний розмір кластеру.

Для того щоб виконати перестановку, необхідно перемістити лише k кластерів, які можуть замістити собою ще k кластерів. Дані кластери можуть розміщуватися по усій довжині матриці стану, тобто переміщення зчитуючої головки для переміщення одного кластеру може бути із крайньої лівої позиції до крайньої правої, тобто необхідно пройтись по усіх кластерах – n. Отже, кількість переміщень зчитуючої головки дорівнює – $O\left(f\left(2kn\right)\right)$. Причому, якщо розмір повідомлення у стеганоблоках наближається до кількості кластерів, то обчис-

лювальна складність переміщень зчитуючої головки наближається до квадратичної залежності – $O\left(f\left(n^2\right)\right); k \to n$.

Кількість зчитувань та записів кластерів фіксована та залежить лише від кількості стеганоблоків, але таких записів може бути по два, коли інформаційний кластер заміщує неінформаційній. Але можливі випадки, коли деякі кластері можна не перезаписувати, як описано вище. А отже $O\left(g\left(2k\right)\right)^{-}$ та $O\left(h\left(2k\right)\right)^{-}$. Результуюча обчислювальна складність матиме вигляд

$$O(B) = O\left(f\left(2kn\right)\right) + O\left(g\left(2k\right)\right)^{-} + O\left(h\left(2k\right)\right)^{-}; n = C_{len}; k = M_{len} \quad (5)$$

Так як не виконується подальше перерозміщення неінформативних кластерів у суцільні послідовності, то, зазвичай, рівень фрагментації покриваючих файлів при перерозміщенні у такий спосіб буде вищий, аніж при виконанні методу приховування інформації шляхом ПчЗОП-І, та через можливу переплетеність деяких файлів рівень фрагментації буде вищий за рівень фрагментації при виконанні приховування повідомлення шляхом ПчЗОП-ІІ. Але для систем, де рівень фрагментації покриваючих файлів не є значним показником, то приховування повідомлення шляхом перемішування кластерів покриваючих файлів спосіб виконання перестановки із почерговим завантаженням кластерів до оперативної пам'яті із оптимальним переміщенням лише інформативних кластерів є оптимальним. Для даного прикладу матимемо показники, що зазначені у табл. 4.

Таблиця 4.

Показники результату приховування прикладу шляхом ПчЗОП-ІІІ

π	(4,11)(5,17)(6,22)(7)(12)(16)(18)(23)(24)
$O\left(f\left(n^2\right)\right)$	36
$O\left(g\left(2k\right)\right)^{-}$	6
$O\left(h\left(2k\right)\right)^{-}$	6
$O\left(m\left(const\right)\right)$	2
$\varphi\left(F_0\right)$	4 (8)
$\varphi\left(F_1\right)$	2 (4)
$\varphi\left(F_2\right)$	2 (6)
$\varphi\left(F_3\right)$	2 (7)

Висновки

Порівняємо дані способи реалізації базового методу приховування інформації шляхом перемішування кластерів покриваючих файлів за такими показниками:

– обчислювальна складність за необхідним об'ємом вільного місця у оперативній пам'яті – $O(m)$;

– обчислювальна складність за часовими показниками – $O(f)$, $O(g)$, $O(h)$;

– захищеність від детектування, тобто за впливом на рівень фрагментації.

Порівнюючи способи реалізації за обчислювальною складністю за часовими показниками, необхідно зазначити, що переміщення зчитуючої головки пристрою займає найменше часу у абсолютних величинах. А запис кластеру займає найбільше часу. Тобто, затрачений час підпорядковується такій закономірності:

$$O(f) < O(g) < O(h) \quad (6)$$

Причому, для SSD технології час, затрачений на переміщення зчитуючої головки, наближається до нуля, так як у SSD пристроях реалізована паралельна обробка секторів, та таке визначення, як зчитуюча головка пристрою, не має сенсу. Отже, результат порівняльного аналізу способів реалізації методу зазначено у табл. 5.

Таблиця 5

Результат порівняльного аналізу способів реалізації базового методу приховування інформації

Спосіб реалізації	Необхідний об'єм ОП	Необхідний час	Захищеність від детектування
ПЗОП	--	-	++
ПчЗОП-I	++	--	++
ПчЗОП-II	++	+	+
ПчЗОП-III	++	++	-

Також відповідні результати отримано і для модифікованого методу приховування даних у структуру файлової системи за способами приховування ПЗОПм, ПчЗОП-I/II/IIIм (дані способи відповідають описаним вище способам використання базового методу).

Роблячи висновок за результатами порівняльного аналізу, можна стверджувати:

– для систем, де розмір оперативної пам'яті є достатнім (розмір покриваючих файлів цілком уміщається у оперативну пам'ять), переважним способом буде ПЗОП. Даний спосіб дозволяє досягти мінімального впливу на рівень фрагментації покриваючих файлів, але ПЗОП потребує значного часу на виконання приховування повідомлення;

– для систем, де затрачений час є критичним параметром, рекомендується використовувати способи ПчЗОП-II та ПчЗОП-III. ПчЗОП-III потребує менше часу на приховування повідомлення, але вплив на рівень фрагментації у такий спосіб найбільший;

– для систем, у яких захищеність від детектування є критичним параметром, рекомендовано використовувати способи ПЗОП та ПчЗОП-I (у залежності від доступного об'єму ОП).

Окремо можна виділити ПчЗОП-II як найоптимальніший спосіб через те, що даний спосіб дозволяє досягти задовільних показників при приховуванні повідомлення без збитку у інших показниках.

Також необхідно зазначити, якщо для приховування повідомлення буде задіяно значну кількість кластерів покриваючих файлів (тобто кількість стеганоблоків до кількості кластерів), то часові переваги способів ПчЗОП-II та ПчЗОП-III нівелюються. У такому випадку бажано використовувати ПЗОП та ПчЗОП-I у залежності від допустимого об'єму ОП.

Відповідно до отриманих досліджень було розроблено програмну реалізацію симуляції (https://github.com/ShekhaninKyryl/SteganoSimulation), яка дозволяє емпіричним шляхом оцінити обчислювальну складність методів приховування.

Узагальнюючи результати, отримані емпіричним шляхом, можна зробити наступний висновок. Для способів приховування інформації базовим методом емпірично отримана обчислювальна складність відповідає теоретично розрахованій. Особливо необхідно виділити способи ПчЗОП-II та ПчЗОП-III, так як кількість перезаписаних кластерів значно нижча, ніж при використанні ПЗОП та ПчЗОП-I. Але з іншого боку, практично отримана обчислювальна складність способів приховування інформації модифікованим методом неповністю відповідає теоретично розрахованій. А саме при використанні ПчЗОП-II/IIIм кількість зчитаних та записаних кластерів залежить від кількості кластерів покриваючих файлів – n, а не від кількості стеганоблоків – k. Частково це можна вирішити, імплементувавши вдосконалену функцію приховування модифікованої компоненти стеганоблоку. Для цього при розрахунку та перемішуванні кластерів у відповідності до базової компоненти треба помічати кластери, які не були переміщені, та за можливістю зберігати їх позиції при виконанні перемішування кластерів за допомогою модифікованої компоненти. Також необхідно виконувати такий

алгоритм рекурсивно – якщо на поточній ітерації приховати необхідне повідомлення неможливо із збереженням позицій усіх кластерів, які необхідно зберегти, то необхідно знехтувати одним таким кластером та знову спробувати приховати повідомлення. І так далі, до поки не вийде приховати усю необхідно інформацію. Узагальнююча оцінка обчислювальної складності показана у табл. 6.

Таблиця 6

Порівняння теоретично розрахованої обчислювальної складності із емпірично отриманою при використанні різних способів

Параметр ОС (теоретичне / емпіричне)	$O(f)$	$O(g)$	$O(h)$	$O(m)$
ПЗОП	$O(f(n))$ / $O(f(n))$	$O(g(n))$ / $O(g(n))$	$O(h(n))$ / $O(h(n))$	$O(m(n))$ / $O(m(n))$
ПчЗОП-I	$O(f(n^2))$ / $O(f(n^2))$	$O(g(n))$ / $O(g(n))$	$O(h(n))$ / $O(h(n))$	$O(m(2)$ / $O(m(2)$
ПчЗОП-II	$O(f(n^2))$ / $O(f(n^2))$	$O(g(2k))$ / $O(g(2k))$	$O(g(2k))$ / $O(g(2k))$	$O(m(2))$ / $O(m(2))$
ПчЗОП-III	$O(f(n^2))$ / $O(f(n^2))$	$O(g(2k))^-$ / $O(g(2k))^-$	$O(h(2k))^-$ / $O(h(2k))^-$	$O(m(2))$ / $O(m(2))$
ПЗОПм	$O(f(n))$ / $O(f(n))$	$O(g(n))$ / $O(g(n))$	$O(h(n))$ / $O(h(n))$	$O(m(n))$ / $O(m(n))$
ПчЗОП-Ім	$O(f(n^2))$ / $O(f(n^2))$	$O(g(n))$ / $O(g(n))$	$O(h(n))$ / $O(h(n))$	$O(m(2)$ / $O(m(2)$
ПчЗОП-IIм	$O(f(n^2))$ / $O(f(n^2))$	$O(g(2k))$ / $O(g(n))$	$O(g(2k))$ / $O(g(n))$	$O(m(2))$ / $O(m(2))$
ПчЗОП-IIIм	$O(f(n^2))$ / $O(f(n^2))$	$O(g(2k))^-$ / $O(g(n))$	$O(h(2k))^-$ / $O(g(n))$	$O(m(2))$ / $O(m(2))$

Список літератури:

1. Kuznetsov A. and others. Method of 3D-steganography // CS&CS E-journal. 2018. № 4. С. 4–12.
2. Kuznetsov A. and others. Information Hiding Using 3D-Printing Technology // 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). 2019a. С. 701–706.
3. Kuznetsov A. A. and others. 3D STEGANOGRAPHY INFORMATION HIDING // TRE. 2019b. Т. 78. № 12.
4. Mazurczyk W., Szczypiorski K. Steganography of VoIP Streams // arXiv:0805.2938 [cs]. 2008.
5. Fraczek W., Mazurczyk W., Szczypiorski K. Stream Control Transmission Protocol Steganography // 2010 International Conference on Multimedia Information Networking and Security. 2010. С. 829–834.
6. Fraczek W., Mazurczyk W., Szczypiorski K. How Hidden Can be Even More Hidden? // 2011 Third International Conference on Multimedia Information Networking and Security, 2011. С. 581–585.
7. Szczypiorski K. HICCUPS: Hidden communication system for corrupted networks // 2003.
8. Khan H. and others. Designing a cluster-based covert channel to evade disk investigation and forensics // Computers & Security. 2011. Т. 30. № 1. С. 35–49.
9. Khan H. and others. Evading Disk Investigation and Forensics using a Cluster-Based Covert Channel // 2012.

10. Venčkauskas A. and others. Covert Channel for Cluster-based File Systems Using Multiple Cover Files // Information Technology and Control. 2013. Т. 42. № 3. С. 260-267.

11. Shekhanin K.Yu., Kolhatin A.O., Demenko E.E., Kuznetsov A.A. On hiding data into the structure of the FAT family file systemy. Telecommunications and Radio Engineering (English translation of Elektrosvyaz and Radiotekhnika): Volume 78, Issue 11, 2019, Pages 973-985. https://www.scopus.com/record/display.uri?eid=2-s2.0-85070406462&origin=inward.

12. Shekhanin K., Kuznetsov A., Krasnobayev V., Smirnov, O: Detecting hidden information in FAT // International Journal of Computer Network and Information Security: Vol. 12, Issue 3, June 2020. P. 33-43. https://www.scopus.com/record/display.uri?eid=2-s2.0-85086029655&origin=inward

13. Kuznetsov A., Shekhanin K., Kolhatin, A., Mikheev I., Belozertsev I. Hiding data in the structure of the FAT family file system // Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT 2018. 9 July 2018, Pages 337-342 https://www.scopus.com/record/display.uri?eid=2-s2.0-85050684345&origin=inward

14. Shekhanin K., Kolhatin A., Kuznetsova K., Kavun S. Steganographic hiding information in a file system structure // 2018 International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2018 // Proceeding. September 2018, P. 9047551. https://www.scopus.com/record/display.uri?eid=2-s2.0-85083488842&origin=inward

Надійшла до редколегії 15.09.2021

Відомості про авторів:

Кузнецов Олександр Олександрович – д-р техн. наук, професор, Харківський національний університет імені В.Н. Каразіна, кафедра безпеки інформаційних систем і технологій, факультет комп'ютерних наук; Україна; e-mail: kuznetsov@karazin.ua, ORCID:https://orcid.org/0000-0003-2331-6326

Шеханін Кирил Юрійович – аспірант, Харківський національний університет імені В.Н. Каразіна, кафедра безпеки інформаційних систем і технологій, факультет комп'ютерних наук; Україна; e-mail: kyryl.shekhanin@karazin.ua, ORCID: https://orcid.org/0000-0002-1441-7814

Пшенична Світлана Вікторівна – старший науковий співробітник, Харківський національний університет імені В.Н. Каразіна, факультет радіофізики, біомедичної електроніки та комп`ютерних систем; Україна; e-mail: kyryl.shekhanin@karazin.ua, ORCID: https://orcid.org/0000-0002-6212-7280