

# ПОРІВНЯЛЬНИЙ АНАЛІЗ ТА ДОСЛІДЖЕННЯ ВЛАСТИВОСТЕЙ НОСІЇВ ІНФОРМАЦІЇ ДЛЯ СТЕГАНОГРАФІЧНОГО ПРИХОВУВАННЯ ДАНИХ В КЛАСТЕРНИХ ФАЙЛОВИХ СИСТЕМАХ

Кирил Шеханін, Людмила Горбачова, Катерина Кузнецова

Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна  
[lusyag23@gmail.com](mailto:lusyag23@gmail.com), [kate7smith12@gmail.com](mailto:kate7smith12@gmail.com)

Рецензент: Микола Карпінський, д.т.н., проф., університет Бельсько-Бяла, Бельсько-Бяла, Польща.  
[mkarpinski@ath.bielsko.pl](mailto:mkarpinski@ath.bielsko.pl)

Надійшла: Грудень 2020.

**Анотація:** У роботі розглянуто сучасні технології збереження інформації, а саме HDD, Flash-USB, SSD. Проаналізовано такі показники, як кількість реалізованих виробів, ціна, швидкість зчитування та запису. Досліджено деякі показники ефективності носіїв інформації, з точки зору можливості застосування стеганографічних методів приховування інформації у кластерних файлових системах. Виконано аналіз швидкості послідовного зчитування/запису та швидкості доступу до випадкового кластеру, що відповідає швидкості доступу до фрагментованого файлу. Для цього використовувалися результати тестувань з ресурсу UserBenchmark. Тестування виконувалися методами Sequential та Random4k. Запропонована оцінка носіїв інформації і надані рекомендації, стосовно використання носія інформації та методу приховування даних шляхом перемішування кластерів у структурі файлової системи. Проведені аналіз взаємозв'язку і залежності параметрів швидкості доступу до кластеру, та рівня фрагментованості файлу. Уточнено, яким чином збільшення чи зменшення рівня фрагментованості (переплетеності) впливає на швидкість доступу до файлу, що є доволі важливим показником при використанні методу приховування даних у структурі файлової системи. Розглянуті переваги і недоліки накопичувачів інформації різних типів, та проведено їх порівняльний аналіз. Виконано, аналіз особливостей процесу дефрагментації накопичувачів, та вплив різних факторів на загальний рівень фрагментованості на носії інформації. Підкреслено, що чим більший рівень фрагментованості на носії даних, тим більше інформації можна приховати. Стверджується, що завдяки широкому розповсюдженню SSD/HDD накопичувачів, метод приховування інформації у структуру файлових систем, шляхом перемішування кластерів покриваючих файлів, є актуальним.

**Ключові слова:** файлові носії інформації; фрагментація; швидкість доступу; приховування даних; стеганографія.

## 1 Вступ

Із розвитком інформаційних технологій питання зберігання, обміну і обробки інформації, як і раніше, не втрачають своєї актуальності. При цьому на передній план виходять напрямки, що пов'язані з процесами зменшення експлуатаційних витрат, підвищення швидкодії та якості відповідних онлайн сервісів та служб. Як свідчить практика, в багатьох випадках інформація повинна бути захищена різними криптографічними методами, а доступ до неї забезпечуватися, тільки авторизованим користувачам. Але у разі використання криптографічних методів потенційним зловмисникам відомо про факт збереження або передачі інформації [1–3]. Це, у подальшому, може призвести до розкриття змісту чутливої інформації шляхом використання методів криптоаналізу. Суттєво ускладнити потенційним зловмисникам процес доступу чутливих даних можна за рахунок додаткового використання різних стеганографічних методів [4–6]. Використання стеганографії доповнює криптографічні методи, тож ці методи є компліментарними один до одного з точки зору захищеності інформації. В цьому сенсі, як приклад, можна згадати сумісне використання шифру Цезаря (криптографічний метод) і «невидимих» чорнил (стеганографічний метод) [4, 5, 7]. А отже, якщо сумісно використати ці два методи, то зловмиснику необхідно буде спочатку визначити і підтвердити факт використання прихованого тексту, що написаний невидимими чорнилами, і тільки вже потім зробити спробу безпосередньо дешифрувати використовуваний шифр.

Звісно у сучасності данні методи вже не є ефективними, а більшої популярності набули методи комп'ютерної, або цифрової стеганографії [4, 5, 7–10], де інформаційні повідомлення подаються у вигляді цифрових даних, що приховуються в інших цифрових даних. Так звані

покрівельні файли, які маскують факт інкапсуляції прихованого вмісту, передаються та зберігаються у відкритому вигляді. Це можуть бути, наприклад, цифрові зображення, які у великій кількості передаються сучасними каналами електронної пошти. При цьому вповноважений отримувач, який має секретний ключ дешифрування, приймає покрівельний файл та може відновити таємне повідомлення. Звісно, що при цьому приховується сам факт існування прихованого повідомлення, а відслідкувати та дослідити всі покрівельні (*тобто, маскуючі*) файли у в'єсет'їнної мережі фізично неможливо [9–11]. Нарізі стеганографічні методи постійно вдосконалюються та розвиваються. Основна вимога до покрівельних файлів є їхня надмірність (збитковість). Наприклад, надмірність цифрових зображень дозволяє приховати досить великі за обсягом інформаційні повідомлення.

Останніми роками набули розвитку методи технічної стеганографії. В таких системах приховування інформації досягається шляхом використання властивостей, які штучно зроблено людиною при побудові різних технічних засобів. Як приклад, можна навести мережеву стеганографію [12–16], і якій застосовуються різні особливості побудови сучасних телекомунікаційних систем та мереж, в тому числі штучна надмірність при визначенні форматів пакетів даних на способів їхньої передачі [17–19]. В 3D стеганографії використовуються надмірності цифрових 3D моделей та створених за їх допомогою фізичних об'єктів [20–24]. Наприклад, у роботах [25, 26] запропоновано приховувати інформацію шляхом створення фізичних об'єктів всередині інших об'єктів.

Ще одним прикладом технічної стеганографії є застосування особливостей побудови кластерних файлових систем. Зокрема, у роботах [27–29] було запропоновано методи, які дозволяють ефективно приховувати інформацію шляхом зміни чергування окремих кластерів т.з. покрівельних файлів. Імена (назви) таких файлів є ключовою інформацією і відновити приховуване повідомлення без посилань (*тобто без назв*) покрівельних файлів вкрай важко. Подальші дослідження кластерних файлових систем [30–36] дозволяють впевнено стверджувати, що застосування особливостей організації зберігання інформації дозволяє отримати надійний та безпечний механізм стеганографічного приховування інформаційних повідомлень. Звісно, що стійкість та швидкодія кластерних стеганосистем безпосередньо спирається на конкретні властивості файлової системи. Зокрема, безпека спирається на фрагментарність та переплетеність (в термінах робіт [27, 27, 30, 33]) покрівельних файлів, а швидкодія залежить від фрагментарності файлової системи, та характеристик конкретного носія інформації. Отже актуальним є питання аналізу різних носіїв інформації і відповідних файлових систем на предмет їх можливого застосування в стеганографічних методах приховування, та дослідження впливу окремих показників на ефективність кластерних стеганосистем.

Метою даної роботи є аналіз відомих технологій та методів зберігання інформації та дослідження властивостей фрагментарності файлової системи і переплетеності окремих файлів, на швидкість процесів запису/зчитування даних. Практичне значення роботи обумовлено визначенням можливих перспектив використання досліджених процедур для цілей побудови кластерних стеганосистем.

## 2 Техніки приховування інформації у кластерні файлові системи

Файлова система встановлює порядок, спосіб організації, зберігання та іменування даних на носіях інформації в інформаційних системах, а також в іншому електронному обладнанні: цифрових фотоапаратах, мобільних телефонах тощо [37–39]. Файлова система визначає формат вмісту і спосіб фізичного зберігання інформації, яка групується у файли. Конкретна файлова система визначає розмір імен файлів і (каталогів), тах можливий розмір файлу і розділу, та атрибути файлу, *тобто визначає метадані файлів*. Деякі файлові системи надають сервісні можливості, наприклад, розмежування доступу або шифрування файлів.

Найпростіші методи приховування інформації у структуру файлових систем розглянуто в [32, 34–36]. Дані методи застосовують вільні кластери (*службові поля даних*) для запису прихованої інформації, але такий спосіб, з точки зору конфіденційності інформації, є ненадійним [27, 29]. Інші методи, наприклад [27–31], ґрунтуються на використанні покриваючих

файлів і приховуванні інформаційних даних за допомогою взаємного перемішування файлів, тобто зміни відносних позицій кластерів декількох покриваючих файлів один до одного.

Приховування інформації через перемішування кластерів різних покриваючих файлів.

Приховування інформації через перемішування кластерів різних покриваючих файлів досліджено у роботах [27, 29]. Приховувана інформація представляється у вигляді бітового масиву:  $M = \{b_0, b_1, \dots, b_{n-1}\}$ ,  $b_i \in \{0, 1\}$ . На інформаційному носії обирають  $p = 2^m$ ,  $m \in \mathbb{N}$  покриваючих файлів:  $F_0, F_1, \dots, F_{p-1}$ . Порядок кластерів покриваючих файлів приховує інформаційне повідомлення, тобто після вбудовування покриваючих файлів не можна модифікувати, видаляти та переміщувати. Натуральне число  $m$  та імена покриваючих файлів є секретним ключем. Важливим є, також, порядок упорядкування покриваючих файлів [29].

Формується масив номерів кластерів покриваючих файлів:

$$C = \begin{pmatrix} c_{0,0} & c_{0,1} & \dots & c_{0,L_0-1} & & & & \\ c_{1,0} & c_{1,1} & \dots & \dots & \dots & \dots & c_{1,L_1-1} & \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \\ c_{p-1,0} & c_{p-1,1} & \dots & \dots & c_{p-1,L_{p-1}-1} & & & \end{pmatrix},$$

де кожен рядок масиву містить номери кластерів відповідного файлу. Наприклад, файлу  $F_i$  відповідає  $i$ -й рядок масиву  $C$ , тобто номери кластерів  $i$ -го покриваючого файлу можуть бути подані у вигляді масиву  $C_{F_i} = \{c_{i,0}, c_{i,1}, \dots, c_{i,L_i-1}\}$ , де  $L_i$  - число кластерів в  $i$ -у покриваючому файлі. Якщо при приховуванні інформації необхідно зберегти без зміни вміст покриваючих файлів, тоді потрібно, щоб виконувалися умови:  $\forall i: L_i \geq k$ ,  $k = n/m$ .

Формується масив  $D$  номерів вільних кластерів файлової системи:  $D = \{c_1, c_2, \dots, c_{L_D}\}$ , так щоб виконувалася умова  $c_1 < c_2 < \dots < c_{L_D}$ . Число  $L_D$  дорівнює кількості вільних кластерів файлової системи, причому потрібно, щоб виконувалася умова:  $L_D \geq \sum_{i=0}^{p-1} L_i$ .

Інформаційне повідомлення  $M$  розбивається на блоки по  $m$  бітів:  $M = \{B_1, B_2, \dots, B_k\}$ , де  $k = \lceil n/m \rceil$  та якщо  $k = n/m$ , то  $B_1 = \{b_0, b_1, \dots, b_{m-1}\}$ ,  $B_2 = \{b_m, b_{m+1}, \dots, b_{2m-1}\}$ ,  $\dots$ ,  $B_k = \{b_{(k-1)m}, b_{(k-1)(m+1)}, \dots, b_{km-1}\}$ . Якщо  $k < n/m$ , то останній блок необхідно дописати неінформаційними значеннями, наприклад, нулями  $B_1 = \{b_0, b_1, \dots, b_{m-1}\}$ ,  $B_2 = \{b_m, b_{m+1}, \dots, b_{2m-1}\}$ ,  $\dots$ ,  $B_k = \{b_{(k-1)m}, b_{(k-1)(m+1)}, \dots, b_{n-1}, \underbrace{0, 0, \dots, 0}_{km-n}\}$ . Кожен блок  $B_i$ ,  $i = 1, 2, \dots, k$  подається як натуральне число, тобто  $\forall i: 0 \leq B_i \leq p-1$ . Кожне натуральне число  $B_i$ ,  $i = 1, 2, \dots, k$  подається як номер покриваючого файлу з множини файлів  $F_0, F_1, \dots, F_{p-1}$ .

Всі кластери покриваючих файлів перезаписуються у вільні кластери файлової системи, тобто масив  $D$  заповнюється номерами кластерів із масиву  $C$ . Порядок перезапису кластерів покриваючих файлів відповідає послідовності натуральних чисел  $\{B_1, B_2, \dots, B_k\}$ , що задаються повідомленням, яке приховується. Наприклад, у перший вільний кластер записуємо 1-й кластер покриваючого файлу із номером  $B_1$ , у 2-й вільний кластер – наступний кластер покриваючого файлу із номером  $B_2$  і так далі.

Натуральні числа  $B_i$  можуть співпадати і в цьому разі записуємо наступний кластер того ж самого покриваючого файлу із номером  $B_i$ . Для посилення захисту від детектування та декодування стеганосистеми додатково можуть застосовуватися певні механізми, наприклад [29], обирається ключ ініціалізації  $B_0$ , а порядок перезапису кластерів покриваючих файлів задається послідовністю натуральних чисел  $\{N_1, N_2, \dots, N_k\}$ ,  $N_i = B_{i-1} + B_i \bmod p$ ,  $0 \leq N_i \leq p-1$ . Тоді, у 1-й вільний кластер перезаписуються 1-й кластер покриваючого файлу із номером  $N_1$ , а у 2-й вільним кластер – черговий кластер покриваючого файлу із номером  $N_2$  і т.д.

В результаті виконання алгоритму перші  $k$  вільних кластерів файлової системи будуть записані кластерами покриваючих файлів. Отже повинна виконуватися умова  $k \leq L_D$ .

Для вилучення прихованого повідомлення  $M$  формується масив  $D$  номерів кластерів покриваючих файлів:  $D = \{c_1, c_2, \dots, c_{L_D}\}$ , причому необхідно, щоб виконувалась умова  $c_1 < c_2 < \dots < c_{L_D}$ . Кожен номер кластеру з цього масиву співвідноситься тільки з одним кластером одного покриваючого файлу. Саме така відповідність пов'язана із логікою вбудовування інформації і використовується для вилучення прихованої інформації. Формується послідовність натуральних чисел  $\{B_1, B_2, \dots, B_k\}$ , які відповідають блокам прихованого повідомлення:  $B_1 = \{b_0, b_1, \dots, b_{m-1}\}$ ,  $B_2 = \{b_m, b_{m+1}, \dots, b_{2m-1}\}$ , ...,  $B_k = \{b_{(k-1)m}, b_{(k-1)(m+1)}, \dots, b_{km-1}\}$ . З цих бітових блоків формується інформаційне повідомлення  $M = \{b_0, b_1, \dots, b_{n-1}\}$ ,  $b_i \in \{0,1\}$ . Якщо  $k < n/m$ , тоді останній блок «обрізується» - його останні  $km - n$  біт не несуть інформаційного значення.

Недоліком даного методу є незначний обсяг розміру прихованої інформації, який залежить від кількості покриваючих файлів та розміру покриваючих файлів у кластерах. Так, у кожному кластері покриваючих файлів може бути приховано  $\log_2 p = m$  інформаційних біт.

Подальший розвиток розглянутого методу наведено у роботах [30, 31], де для збільшення обсягу прихованого повідомлення, в модифікованому методі, запропоновано додаткове перемішування кластерів кожного покриваючого файлу.

Модифікований метод приховування інформації в кластерних файлових системах ґрунтується на використанні одного або декількох покриваючих файлів і приховуванні інформаційного повідомлення шляхом зміни порядку позицій кластерів різних покриваючих файлів, та зміни чергування кластерів у межах одного покриваючого файлу. Даний метод дозволяє досягти збільшення прихованої інформації на один біт для кожного кластеру, при одних і тих значеннях ключових параметрів. Приховування інформації подається у вигляді бітового масиву:  $M = \{b_0^*, b_1^*, \dots, b_{L_1+L_2+\dots+L_{p-1}-1}^*, b_0, b_1, \dots, b_{n-1}\}$ ,  $b_i^*, b_i \in \{0,1\}$ . На носії інформації обирається  $p = 2^m$ ,  $m \in \mathbb{N}$  покриваючих файлів:  $F_0, F_1, \dots, F_{p-1}$ . Після цього формується масив номерів кластерів покриваючих файлів, як в вище описаному методі.

В даному разі для кожного покриваючого файлу змінюється порядок «чергування» кластерів у кожному покриваючому файлі. Порядок «чергування» задається відповідною послідовністю  $M$ . Для цього, формується  $p$  бітових масивів інформаційних бітів:

$$\begin{aligned} M_1 &= \{b_0^*, b_1^*, \dots, b_{L_1-1}^*\}, \\ M_2 &= \{b_{L_1}^*, b_{L_1+1}^*, \dots, b_{L_1+L_2-1}^*\}, \\ &\dots \\ M_{L_p} &= \{b_{L_1+L_2+\dots+L_{p-1}}^*, b_{L_1+L_2+\dots+L_{p-1}+1}^*, \dots, b_{L_1+L_2+\dots+L_{p-1}-1}^*\}, \end{aligned}$$

кожен з яких, зіставляється із масивом номерів кластерів покриваючих файлів

$$\begin{aligned} C_{F_1} &= \{c_{1,0}, c_{1,1}, \dots, c_{1,L_1-1}\}, \\ C_{F_2} &= \{c_{2,0}, c_{2,1}, \dots, c_{2,L_2-1}\}, \\ &\dots \\ C_{F_p} &= \{c_{p,0}, c_{p,1}, \dots, c_{p,L_p-1}\}. \end{aligned}$$

Позиції кластерів кожного покриваючого файлу перезаписуються, тобто номери кластерів у кожному з масивів  $C_{F_1}, C_{F_2}, \dots, C_{F_p}$  змінюють своє чергування, відповідно до значень бітових масивів  $M_1, M_2, \dots, M_{L_p}$ . У результаті отримують нові масиви номерів кластерів  $C_{F_1}^*, C_{F_2}^*, \dots, C_{F_p}^*$ .

Перезапис позицій кластерів у кожному покриваючому файлі може здійснюватися різними способами. Наприклад, шляхом розбиття всіх номерів позицій кластерів  $\{c_{i,0}, c_{i,1}, \dots, c_{i,L_i-1}\}$  на дві половини, та співставлення кожної половини із значенням інформації.

ційного біту. Тоді, наприклад, якщо  $b_j^* = 1$ ,  $L_1 + L_2 + \dots + L_{i-1} - 1 < j \leq L_1 + L_2 + \dots + L_i - 1$ , на  $j$ -у позицію в масиві  $C_{F_i}^*$  розміщують кластер з 1-ої половини впорядкованих номерів, якщо  $b_0^* = 0$  - з другої половини номерів.

Сформовані таким чином масиви  $C_{F_i}^* = \{c_{i,0}^*, c_{i,1}^*, \dots, c_{i,L_i-1}^*\}$  перезаписаних позицій номерів покриваючих файлів утворюють масив

$$C^* = \begin{pmatrix} c_{0,0}^* & c_{0,1}^* & \dots & c_{0,L_0-1}^* & & & & \\ c_{1,0}^* & c_{1,1}^* & \dots & \dots & \dots & \dots & \dots & c_{1,L_1-1}^* \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p-1,0}^* & c_{p-1,1}^* & \dots & \dots & \dots & \dots & c_{p-1,L_{p-1}-1}^* & \dots \end{pmatrix}.$$

Зміна порядку чергування кластерів в кожному покриваючому фалі дозволяє приховати перші  $L_1 + L_2 + \dots + L_{p-1}$  інформаційних бітів з масиву  $M$ , тобто інформаційну послідовність  $\{b_0^*, b_1^*, \dots, b_{L_1+L_2+\dots+L_{p-1}-1}^*\}$ . Решту  $n$  інформаційних бітів необхідно приховати так само, як і у розглянутому вище методі [29].

Формується масив  $D$  номерів вільних кластерів файлової системи:  $D = \{c_1, c_2, \dots, c_{L_D}\}$ ,  $c_1 < c_2 < \dots < c_{L_D}$ . Послідовність із інформаційних бітів  $\{b_0, b_1, \dots, b_{n-1}\}$  розбивається на блоки по  $m$  бітів кожен:  $\{B_1, B_2, \dots, B_k\}$ , кожен блок  $B_i$ ,  $i = 1, 2, \dots, k$  подається як натуральне число, тобто  $\forall i: 0 \leq B_i \leq p-1$ . Кожне натуральне число  $B_i$ ,  $i = 1, 2, \dots, k$  подається як номер покриваючого файлу з множини файлів  $F_0, F_1, \dots, F_{p-1}$ .

Номери позицій кластерів покриваючих файлів перезаписуються у вільні кластери, тобто масив  $D$  заповнюється номерами кластерів з масиву  $C^*$  (*перезаписаними кластерами, тобто із зміненим чергуванням кластерів у кожному покриваючому файлі*). При цьому, порядок перезапису кластерів покриваючих файлів відповідає послідовності натуральних чисел  $\{B_1, B_2, \dots, B_k\}$ , які задаються приховуваним повідомленням. Наприклад, у перший порожній кластер перезаписуються 1-й кластер покриваючого файлу із номером  $B_1$ , у другий порожній кластер – черговий кластер покриваючого файлу із номером  $B_2$  і т.д. Натуральні числа  $B_i$  можуть співпадати і в цьому разі записуються чергові кластери того ж самого покриваючого файлу із номером  $B_i$ . В результаті перші  $k$  вільних кластерів файлової системи будуть записані кластерами покриваючих файлів.

Для вилучення прихованого повідомлення  $M$  формується масив  $D$  номерів позицій кластерів покриваючих файлів:  $D = \{c_1, c_2, \dots, c_{L_D}\}$ . Кожен номер кластеру з цього масиву співвідноситься тільки з одним кластером одного покриваючого файлу. При цьому формується послідовність натуральних чисел  $\{B_1, B_2, \dots, B_k\}$ , які відповідають блокам прихованого повідомлення, тобто формується інформаційна послідовність  $\{b_0, b_1, \dots, b_{n-1}\}$ ,  $b_i \in \{0, 1\}$ .

Далі вилучається інформаційна послідовність  $\{b_0^*, b_1^*, \dots, b_{L_1+L_2+\dots+L_{p-1}-1}^*\}$ ,  $b_i^* \in \{0, 1\}$ .

Для цього аналізуємо масиви  $C_{F_i}^* = \{c_{i,0}^*, c_{i,1}^*, \dots, c_{i,L_i-1}^*\}$  номерів кластерів кожного покриваючого файлу. Правило вилучення інформації відповідає логіці приховування. Наприклад, може застосовуватися розбиття всіх впорядкованих позицій номерів  $\{c_{i,0}, c_{i,1}, \dots, c_{i,L_i-1}\}$  на дві половини і співставлення кожної половини із значенням інформаційного біту. Тоді, якщо на  $j$ -й позиції в масиві  $C_{F_i}^*$  розміщено кластер з 1-ї половини масиву впорядкованих номерів  $\{c_{i,0}, c_{i,1}, \dots, c_{i,L_i-1}\}$ , приймають бітове значення  $b_j^* = 1$ . Якщо з другої половини, то приймають бітове значення  $b_0^* = 0$ .

Таким чином, за рахунок додаткової зміни порядку чергування номерів кластерів у кожному із покриваючих файлів вдається збільшити обсяг прихованої інформації. Зокрема, в

порівнянні із способом-прототипом, додатково вдається приховати по одному інформаційному біту на кожен кластер покриваючого файлу.

### 3 Аналіз відомих технологій зберігання інформації

При підготовці матеріалів даної роботи було проаналізовано носії інформації, які використовують різні технології способу збереження інформації:

1. SSD (*Solid-State Drive*) – це енергонезалежний немеханічний запам'ятовуючий пристрій на основі мікросхем пам'яті та керуючому контролері
2. HDD (*Hard (magnetic) Disk Drive*) – це енергонезалежний запам'ятовуючий пристрій оснований на принципі магнітного запису на диск;
3. Flash-USB (*Universal Serial Bus*) – це енергонезалежний запам'ятовуючий пристрій на основі мікросхем, що має структуру подібну до SSD накопичувачів, але має менші показники місткості і швидкодії, тому здебільшого використовуються для перенесення інформації з пристрою на пристрій, та/або відокремленого збереження незначної кількості важливої інформації (паролі, ключі доступу та ін).

Так як Flash-USB є технологією подібною до SSD то переваги й недоліки у даних технології будуть схожі. Але, Flash-USB обмежені максимальним об'ємом зберігаємої інформації, а також інтерфейсом USB (1.5 Мбіт/с ÷ 5 Гбіт/с), що є значно повільнішим за SATA (1.5÷6 Гбіт/с) чи M.2 PCIe (8÷32 Гбіт/с), через які зазвичай підключають SSD накопичувачі до комп'ютерів. В цілому, порівнюючи переваги та недоліки різних SSD та HDD накопичувачів можна стверджувати, що SSD мають більше переваг ніж недоліків (табл. 1).

Таблиця 1 – Порівняння основних параметрів SSD та HDD пристроїв

Архітектура накопичувача	SSD	HDD
Рівень	Майже відсутній	Значний, так як є рухомі частини
Механічна стійкість	Висока	Низька ( <i>при падінні та ударах рухоми частини можуть бути пошкодженні</i> )
Енергоспоживання	2-3 Ватт/годину	5-6 Ватт/годину
Магнітна чутливість	Майже відсутня	Значна ( <i>електромагнітне поле впливає на магнітний диск</i> )
Розміри	Менші розміри та вага	Більші розміри через присутність рухомих частин
Паралельні операції	Підтримуються ( <i>пришвидшує запис/зчитування декількох файлів</i> )	Відсутні
Швидкість запису/зчитування (Мб/с)	1000-3200/2000-4000	100-320/200-400
Ціна (\$/Гб)	0.5	0.1
Циклів перезапису ( <i>разів</i> )	10000	> 100000
Вплив фрагментації на роботу	Майже відсутній ( <i>швидкість зчитування/запису не залежить від фрагментованості файлів</i> )	Значний

Дані цієї таблиці мають дещо приблизний характер через те, що кожен виробник має свої характеристики стосовно пристроїв SSD та HDD. Так, одна й та ж технологія, навіть у одного виробника, може мати різні показники у залежності від цінової категорії пристрою.

Головною метою відомостей табл. 1, було показати *загальну тенденцію* та *різницю* між SSD та HDD накопичувачами.

Варто зазначити, що з розвитком технологій SSD накопичувачі, у порівнянні з аналогами п'ятирічної давнини, мають значний приріст у показниках. У той час як показники HDD пристроїв майже не змінилися. Так, якщо ще у 2015 році кількість HDD пристроїв значно переважала на ринку накопичувачів даних, то вже у 2020 році, кількість пристроїв HDD та SSD практично порівнялась. Зведена гістограма наведена на рис. 1 (*реалізація вироблених накопичувачів інформації по всьому світі, млн. одиниць*).

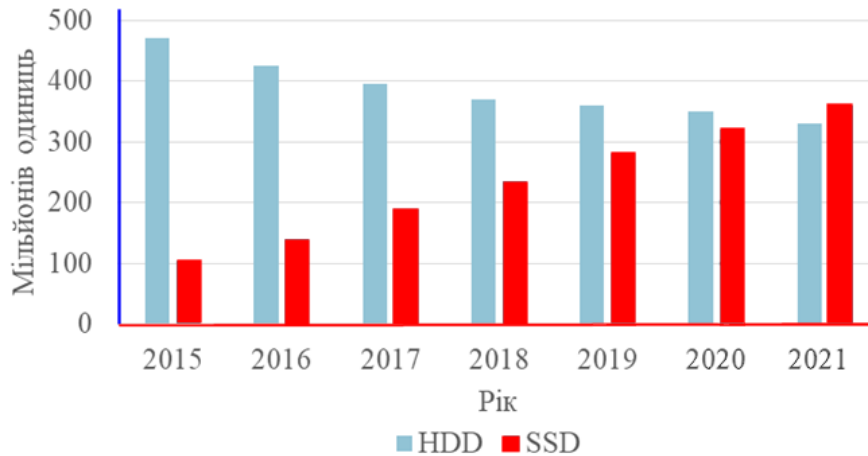


Рис. 1 – Показники реалізації вироблених накопичувачів інформації

Таким чином можна стверджувати, що з кожним роком кількість SSD накопичувачів росте, поглинаючи долю ринку HDD. Динаміка кількості HDD та SSD накопичувачів приведено на рис. 2.

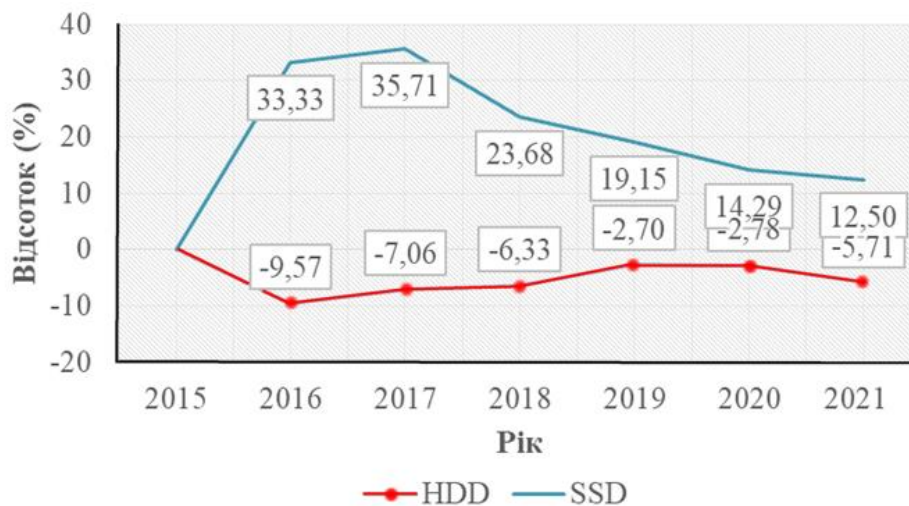


Рис. 2 – Динаміка зміни кількості HDD та SDD пристроїв

Вплив технології накопичувача на можливості методу приховування інформації у структуру файлових систем [31, 33], більш детально оцінимо швидкодію кожної технології. Так як єдиної специфікації по SSD, HDD технологіям не існує, а швидкодія кожного пристрою залежить від виробника та цінової категорії, тож порівняльна оцінка зроблена спираючись на результати електронного ресурсу UserBenchmark (<https://www.userbenchmark.com>).

Даний ресурс збирає дані по компонентам комп'ютера (CPU, GPU, SSD, HDD, RAM, USB) проводячи тестування «Open Source» програмним забезпеченням. Дані з UserBenchmark є об'єктивними так як на ресурсі вказані результати більш ніж 160 млн. користувачів.

#### 4 Методика дослідження та результати експериментів

Слід підкреслити, що в межах виконання даної роботи нас цікавили лише SSD та HDD накопичувачі, так як для методу приховування інформації [31], важливим є параметр швидкості запису/зчитування з носія інформації. Тому, в якості тестових було обрано найкращі пристрої за цими показниками: - для SSD, це – Gigabyte, mod. GP-ASM2NE6100TTTD Aorus NVMe PCIe M.2 1TB (190\$); – для HDD, це – WD, mod. WD6001FZWX Black 6TB (225\$).

Тестування даних пристрої проводилось наступними методами:

1. *Послідовного запису (Sequential)* – це модель доступу до диску, при якій великі блоки даних записуються у сусідні блоки на поверхні пристрою з глибиною черги, рівній одиниці. Даний термін використовується здебільшого у контексті порівняльного аналізу, швидкість вимірюють у МБ/с. Даний тип доступу часто використовують при зчитуванні/запису великих за розміром файлів, таких як відео, музика та зображення. Як демонструє практика, близько 50% звичайного доступу користувачів до дискового накопичувача на їх персональному комп'ютері (ПК), складається з послідовних операцій зчитування/запису. Таки накопичувачі, що використовуються для зберігання великих мультимедійних файлів та/або резервних копій, повинні мати відносно велику швидкість послідовного доступу до файлів;
2. *Випадкового запису 4K (Random 4k)* – це шаблон доступу до диску, при якому невеликі (4k) блоки даних записуються у випадкові місця на поверхні тестового пристрою, з глибиною черги рівній одиниці. Даний термін використовується здебільшого у контексті порівняльного аналізу, швидкість вимірюють у МБ/с. Цій метод оцінки швидкодії можна використовувати для визначення того, наскільки ефективно пристрій зчитує/записує невеликі фрагменти даних із випадкових місць. Даний шаблон доступу характерний під час завантаження операційної системи, коли з накопичувача необхідно зчитати велику кількість файлів конфігурації та драйверів. Як демонструє практика, близько 20% звичайного доступу користувачів до дисків ПК, складається з запису/зчитування, саме з випадкових блоків накопичувача.

Результат тестування *Random 4k* є більш важливим для методу приховування інформації шляхом перемішування кластерів у структурі файлової системи так як при приховуванні даних запис виконується у «випадкові» блоки накопичувача у залежності від приховуваної інформації. Що відповідає швидкості доступу до файлу із значним рівнем фрагментації.

Також необхідно зазначити що модифікований метод приховування даних у структуру файлової системи FAT (який запропоновано у [31]) дозволяє використовувати 1 покриваючий файл з неперервним ланцюгом кластерів але з певним рівнем переплетеності.

Переплетеність – це властивість файлу при якому кластери файлу розміщуються безперервно, без проміжків між кластерами, але можливий зворотній напрямок у індексації кластерів. Як приклад переплетений файл, це файл кластери якого розміщені з 1 по 10 кластери, але ланцюг кластерів може мати вигляд як  $Chain_F = [1, 9, 2, 3, 10, 4, 8, 7, 5, 6]$ . У загальному вигляді вираз, який визначає рівень переплетеності файлу має наступний вигляд (1):

$$Ent_F = \sum_{i=1}^{F_{len}} |Chain_F[i] - Chain_F[i+1]| - 1, \quad (1)$$

де:

- $Ent_F$  (*entanglement*) – рівень переплетеності файлу  $F$ ;
- $F_{len}$  – довжина файлу  $F$  у кластерах;
- $Chain_F[i]$  –  $i$  елемент ланцюгу кластерів файлу  $F$ .

Т.ч., рівень переплетеності файлу у попередньому прикладі буде дорівнювати:

$$Ent_F = (|1-9|-1) + (|9-2|-1) + \dots + (|5-6|-1) = 28.$$



Так як в межах даної роботи виконувався аналіз фізичних властивостей носіїв інформації, то можна прирівняти вплив фрагментованості файлу до впливу переплетеності файлу на швидкість запису/зчитування. Адже, як фрагментованість, так і переплетеність змушують зміщуватися датчик зчитуючого пристрою (для HDD) для доступу до наступного кластеру.

Зведені результати тестувань швидкодії SSD та HDD пристроїв за результатами UserBenchmark наведені у табл. 2 і 3, де: - *T* – технологія пристрою; - *Mt* – метод оцінювання; - *R (read)* – швидкість зчитування даних; - *W (write)* – швидкість запису даних; - *M (mixed)* – швидкість почергового зчитування/запису даних.

Таблиця 2 – Результати UserBenchmark методом Sequential

<b>T</b>	<b>SSD (МБ/с)</b>			<b>HDD (МБ/с)</b>		
<b>Mt</b>	<b>Sequential</b>			<b>Sequential</b>		
	Min.	Avg.	Max.	Min.	Avg.	Max.
<b>R</b>	809	1950	2318	102	167	215
<b>W</b>	1705	3115	3783	138	202	246
<b>M</b>	704	1998	2315	67.5	102	215

Таблиця 3 – Результати UserBenchmark методом Random 4k

<b>T</b>	<b>SSD (МБ/с)</b>		<b>HDD (МБ/с)</b>	
<b>Mt</b>	<b>Random 4k</b>		<b>Random 4k</b>	
	Min.	Avg.	Min.	Avg.
<b>R</b>	33	49.5	R	33
<b>W</b>	103	176	W	103
<b>M</b>	50.7	78.2	M	50.7

Таким чином, якщо порівнювати швидкості накопичувачів, то безпосередньо SSD є більш швидким, у абсолютних значеннях. Якщо порівнювати зниження швидкості між Random 4k і Sequential, то SSD при фрагментованому запису має швидкість у 5-10%, у той час, як HDD втрачають свою швидкість навіть до 1.5-3%. Звісно 5-10% це мало, але це значно краще ніж для випадку використання HDD. Наведемо характерний приклад впливу фрагментованості носія, на швидкість доступу до файлу. Для цього проаналізуємо середній час запису і читання даних з диску при його фрагментації, та без неї. При цьому зауважимо, що середній час читання з диску, це час, який у середньому необхідний на здійснення однієї операції читання даних з носія інформації (*в секундах*). Відповідні відомості вказані на рис. 3 та 4 (*за даними, <https://club.directum.ru/post/140>*).

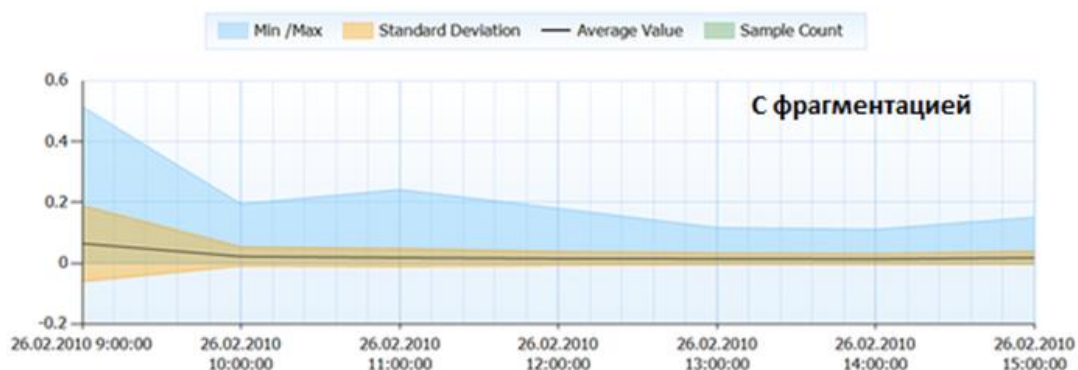


Рис. 3 – Середній час зчитування даних з диску при фрагментації даних

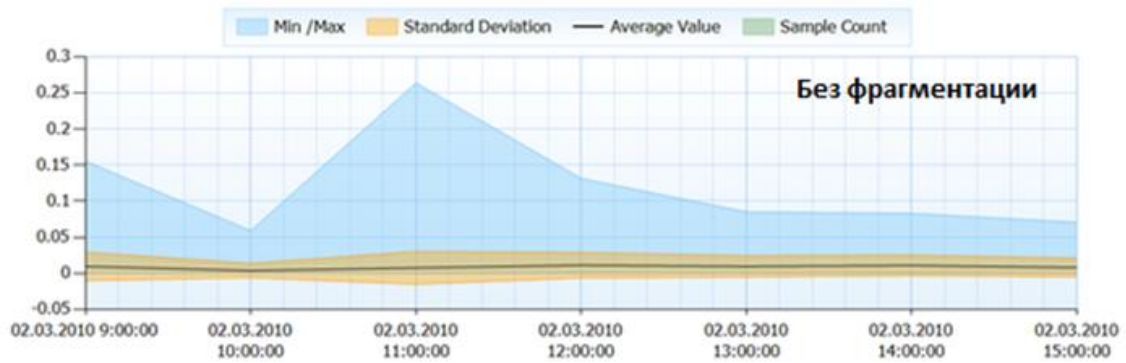


Рис. 4 – Середній час зчитування даних з диску без фрагментації даних

Додаткові результати, також, наведені у таблиці 4.

Таблиця 4 – Оцінка затраченого часу при читанні даних з диску

	Мінімальне значення (с)	Середнє значення (с)	Максимальне значення (с)
З фрагментацією	0	0.23	0.5
Без фрагментації	0	0.08	0.26

Також проаналізуємо середній час запису на диск. Так само оцінюючи час при фрагментованості та без фрагментованості. Результати вказані на рисунку 5 та таблиці 5 (за даними <https://club.directum.ru/post/140>).

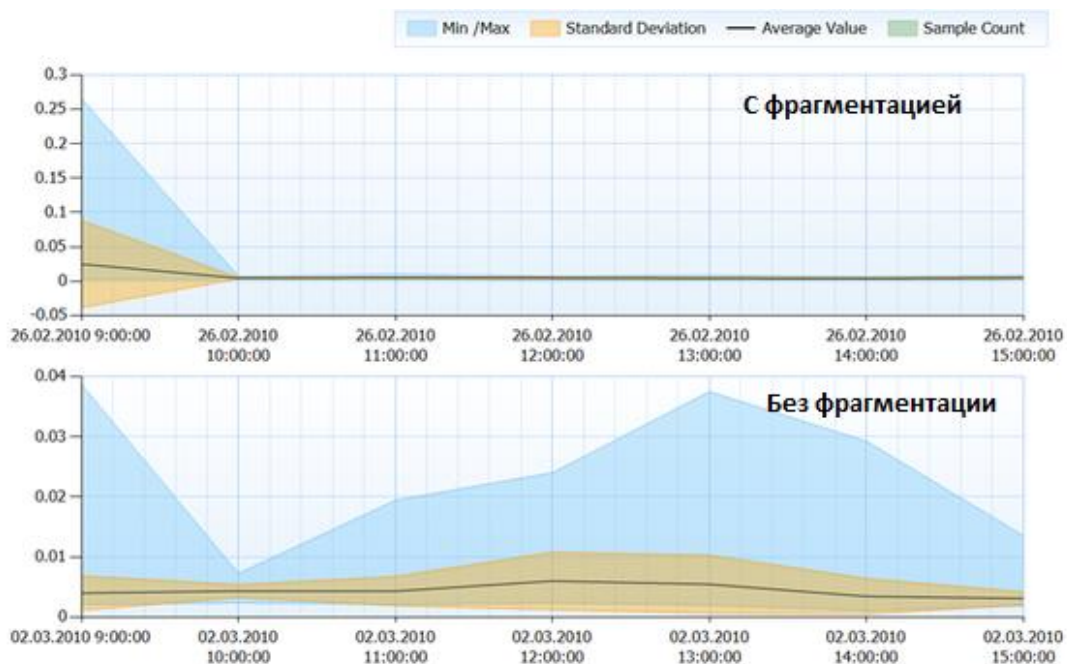


Рис. 5 – Середній час запису даних на носій при фрагментації та без неї

Таблиця 5 – Оцінка затраченого часу при запису даних на диск

	Мінімальне значення (с)	Середнє значення (с)
З фрагментацією	0,001787	0,007211
Без фрагментації	0,001044	0,004368

Порівнюючи результати обробки даних з фрагментацією та без неї, можна стверджувати, що обробка даних без фрагментації ефективніша на 40% - 60%, ніж обробка фрагментованих даних. Але використання SSD здатне зменшити негативний вплив фрагментованості даних на швидкість обробки цих даних. Виходячи з результатів методів *Random 4k* та *Sequential* можна стверджувати, що чим більший рівень фрагментації, тим менша швидкість доступу до файлу. Особливо це впливає на швидкодію HDD накопичувачів, у той час як SSD накопичувачі дозволяють проводити запис/зчитування файли майже без втрати швидкості, навіть при значному рівні фрагментованості даних.

## 5 Висновки

На закінчення можна сказати, що технологія SSD має значні переваги, завдяки чому накопичувачі з даною технологією стають більш розповсюдженими. SSD має у рази більшу швидкість доступу до файлів/секторів, навіть при значному показнику фрагментованості – це сприятливий показник для використання стеганографічних методів приховування даних. Також SSD має обмежену кількість циклів перезапису, і хоча це й недолік, але для методу приховування даних цей показник сприятливий, адже виконання дефрагментації є небажаною операцією для SSD.

Отже як висновок можна рекомендувати використання SSD носіїв інформації для приховування повідомлення у структуру файлової системи:

1 - через те що швидкість доступу до кластеру значно вища, що забезпечить більше швидке виконання стеганографічного методу;

2 - при збільшенні рівня фрагментованості (переплетеності) швидкість доступу до файлу втрачає не так багато як у порівнянні з HDD технологією, що є значно важливішим показником при використанні методу приховування даних у структурі файлової системи;

3 - виконання дефрагментації SSD накопичувачів є небажаною процедурою, що призводить до збільшення загального рівня фрагментованості на носії інформації, що у свою чергу дозволяє приховати більше інформації без ризику розкриття (чим більший рівень фрагментованості на носії тим більше інформації можна приховати [33]).

Таким чином завдяки розповсюдженню SSD метод приховування інформації у структуру файлових систем шляхом перемішування кластерів покриваючих файлів є актуальним.

## Посилання

- [1] Klima, R.E., Klima, R., Sigmon, N.P., Sigmon, N., Klima, R., Sigmon, N.P., Sigmon, N.: *Cryptology : Classical and Modern*. Chapman and Hall/CRC (2018). <https://doi.org/10.1201/9781315170664>.
- [2] Delfs, H., Knebl, H.: *Introduction to Cryptography*. Springer Berlin Heidelberg, Berlin, Heidelberg (2015). <https://doi.org/10.1007/978-3-662-47974-2>.
- [3] Childs, L.N.: *Cryptology and Error Correction: An Algebraic Introduction and Real-World Applications*. Springer International Publishing, Cham (2019). <https://doi.org/10.1007/978-3-030-15453-0>.
- [4] Manoj, I.V.S.: *Cryptography and Steganography*. IJCA. 1, 63–68 (2010). <https://doi.org/10.5120/257-414>.
- [5] Yahya, A.: *Introduction to Steganography*. In: Yahya, A. (ed.) *Steganography Techniques for Digital Images*. pp. 1–7. Springer International Publishing, Cham (2019). [https://doi.org/10.1007/978-3-319-78597-4\\_1](https://doi.org/10.1007/978-3-319-78597-4_1).
- [6] Qin, J., Luo, Y., Xiang, X., Tan, Y., Huang, H.: *Coverless Image Steganography: A Survey*. IEEE Access. 7, 171372–171394 (2019). <https://doi.org/10.1109/ACCESS.2019.2955452>.
- [7] Schöttle, P., Böhme, R.: *Game Theory and Adaptive Steganography*. IEEE Transactions on Information Forensics and Security. 11, 760–773 (2016). <https://doi.org/10.1109/TIFS.2015.2509941>.
- [8] Yahya, A.: *Steganography Techniques*. In: Yahya, A. (ed.) *Steganography Techniques for Digital Images*. pp. 9–42. Springer International Publishing, Cham (2019). [https://doi.org/10.1007/978-3-319-78597-4\\_2](https://doi.org/10.1007/978-3-319-78597-4_2).
- [9] Fridrich, J.: *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, Cambridge ; New York (2009).
- [10] Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: *Digital Watermarking and Steganography*, 2nd Ed. Morgan Kaufmann, Amsterdam ; Boston (2007).
- [11] Kim, C.R., Lee, S.H., Lee, J.H., Park, J.-I.: *Blind decoding of image steganography using entropy model*. Electronics Letters. 54, 626–628 (2018). <https://doi.org/10.1049/el.2017.4276>.
- [12] Rowland, C.H.: *Covert channels in the TCP/IP protocol suite*, <https://firstmonday.org/ojs/index.php/fm/article/download/528/449?inline=1>, Last Accessed 2020/11/01.

- [13] Mazurczyk, W., Lubacz, J.: LACK—a VoIP steganographic method. *Telecommun Syst.* 45, 153–163 (2010). <https://doi.org/10.1007/s11235-009-9245-y>.
- [14] Lubacz, J., Mazurczyk, W., Szczypiorski, K.: Principles and Overview of Network Steganography. *IEEE Communications Magazine.* 52, (2012). <https://doi.org/10.1109/MCOM.2014.6815916>.
- [15] Mazurczyk, W., Smolarczyk, M., Szczypiorski, K.: On information hiding in retransmissions. *Telecommun Syst.* 52, 1113–1121 (2013). <https://doi.org/10.1007/s11235-011-9617-y>.
- [16] Cauich, E., Gómez Cárdenas, R., Watanabe, R.: Data Hiding in Identification and Offset IP Fields. In: Ramos, F.F., Larios Rosillo, V., and Unger, H. (eds.) *Advanced Distributed Systems.* pp. 118–125. Springer, Berlin, Heidelberg (2005). [https://doi.org/10.1007/11533962\\_11](https://doi.org/10.1007/11533962_11).
- [17] Wang, M., Gu, W., Ma, C.: A Multimode Network Steganography for Covert Wireless Communication Based on BitTorrent, <https://www.hindawi.com/journals/scn/2020/8848315/>, last accessed 2020/11/08. <https://doi.org/10.1155/2020/8848315>.
- [18] Seo, J.O., Manoharan, S., Mahanti, A.: Network steganography and steganalysis - a concise review. In: 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT). pp. 368–371 (2016). <https://doi.org/10.1109/ICATCCT.2016.7912025>.
- [19] Noskov, A.: Analysis of Network Protocols: The Ability of Concealing the Information. *Computer and Network Security.* (2020). <https://doi.org/10.5772/intechopen.88098>.
- [20] A High Capacity 3D Steganography Algorithm, <https://www.computer.org/csdl/journal/tg/2009/02/ttg2009020274/13rRUwdIOUD>, Last Accessed 2020/10/20. <https://doi.org/10.1109/TVCG.2008.94>.
- [21] Paramasivan, T., Natarajan, V., Gnanasekaran, A., Venkatesan, V., Anitha, R.: Pattern based 3D image Steganography. *3D Research.* 4, (2014). [https://doi.org/10.1007/3DRes.01\(2013\)1](https://doi.org/10.1007/3DRes.01(2013)1).
- [22] Chao, M.-W., Lin, C., Yu, C.-W., Lee, T.-Y.: A high capacity 3D steganography algorithm. *IEEE Trans Vis Comput Graph.* 15, 274–284 (2009). <https://doi.org/10.1109/TVCG.2008.94>.
- [23] Li, N., Hu, J., Sun, R., Wang, S., Luo, Z.: A High-Capacity 3D Steganography Algorithm With Adjustable Distortion. *IEEE Access.* 5, 24457–24466 (2017). <https://doi.org/10.1109/ACCESS.2017.2767072>.
- [24] Thiagarajan, P., Natarajan, V., Aghila, G., Prasanna Venkatesan, V., Anitha, R.: Pattern based 3D image Steganography. *3D Res.* 4, 1 (2014). [https://doi.org/10.1007/3DRes.01\(2013\)1](https://doi.org/10.1007/3DRes.01(2013)1).
- [25] Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Smirnov, O., Krasnobayev, V., Kuznetsova, K.: Information Hiding Using 3D-Printing Technology. In: 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). pp. 701–706 (2019). <https://doi.org/10.1109/IDAACS.2019.8924352>.
- [26] Kuznetsov, A.A., Stefanovych, O.O., Prokopovych-Tkachenko, D.I., Kuznetsova, K.O.: 3D STEGANOGRAPHY INFORMATION HIDING. *TRE.* 78, (2019). <https://doi.org/10.1615/TelecomRadEng.v78.i12.30>.
- [27] Khan, H., Javed, M., Mirza, F., Khayam, S.: Evading Disk Investigation and Forensics using a Cluster-Based Covert Channel. (2012).
- [28] Khan, H., Javed, M., Khayam, S.A., Mirza, F.: Designing a cluster-based covert channel to evade disk investigation and forensics. *Computers & Security.* 30, 35–49 (2011). <https://doi.org/10.1016/j.cose.2010.10.005>.
- [29] Venčkauskas, A., Morkevičius, N., Petraitis, G., Ceponis, J.: Covert Channel for Cluster-based File Systems Using Multiple Cover Files. *Information technology and control.* 42, (2013). <https://doi.org/10.5755/j01.itc.42.3.3328>.
- [30] Kuznetsov, A., Shekhanin, K., Kolhatin, A., Mikheev, I., Belozertsev, I.: Hiding data in the structure of the FAT family file system. In: 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT). pp. 337–342 (2018). <https://doi.org/10.1109/DESSERT.2018.8409155>.
- [31] Shekhanin, K.Y., Kolhatin, A.O., Demenko, E.E., Kuznetsov, A.A.: ON HIDING DATA INTO THE STRUCTURE OF THE FAT FAMILY FILE SYSTEM. *TRE.* 78, (2019). <https://doi.org/10.1615/TelecomRadEng.v78.i11.50>.
- [32] Vokorokos, L., Madoš, B., Ádám, N., Baláž, A., Porubán, J., Chovancová, E.: Multi-Carrier Steganographic Algorithm Using File Fragmentation of FAT FS. *COMPUTING AND INFORMATICS.* 38, 343–366–366 (2019).
- [33] Shekhanin, K., Kuznetsov, A., Krasnobayev, V., Smirnov, O.: Detecting Hidden Information in FAT. *IJCNIS.* 12, 33–43 (2020). <https://doi.org/10.5815/ijcnis.2020.03.04>.
- [34] Aycock, J., de Castro, D.M.N.: Permutation Steganography in FAT Filesystems. In: Shi, Y.Q. (ed.) *Transactions on Data Hiding and Multimedia Security X.* pp. 92–105. Springer, Berlin, Heidelberg (2015). <https://doi.org/10.1007/978-3-662-46739-86>.
- [35] Davis, J., MacLean, J., Dampier, D.: *Methods of Information Hiding and Detection in File Systems.* (2010). <https://doi.org/10.1109/SADFE.2010.17>.
- [36] Neuner, S., Voyiatzis, A.G., Schmiedecker, M., Brunthaler, S., Katzenbeisser, S., Weippl, E.R.: Time is on my side: Steganography in filesystem metadata. *Digital Investigation.* 18, S76–S86 (2016). <https://doi.org/10.1016/j.diin.2016.04.010>.
- [37] FAT File System, [https://www.keil.com/pack/doc/mw/FileSystem/html/fat\\_fs.html](https://www.keil.com/pack/doc/mw/FileSystem/html/fat_fs.html), Last Accessed 2020/10/01.
- [38] FAT File Systems. FAT32, FAT16, FAT12 - NTFS.com, [https://www.ntfs.com/fat\\_systems.htm](https://www.ntfs.com/fat_systems.htm), Last Accessed 2020/10/10.
- [39] Overview of FAT, HPFS, and NTFS File Systems, <https://support.microsoft.com/en-us/help/100108/overview-of-fat-hpfs-and-ntfs-file-systems>, Last Accessed 2020/10/10.

**Reviewer:** Mikołaj Karpiński, Dr. of Sciences (Eng.), Full Prof., University of Bielsko-Biala, Bielsko-Biala, Poland.  
E-mail: [mkarpinski@ath.bielsko.pl](mailto:mkarpiński@ath.bielsko.pl)

Received on December 2020.

**Authors:**

Kirill Shekhanin, Postgraduate Student, Faculty of Computer Science, V.N.Karazin Kharkiv National University, Kharkov, Ukraine.

Lyudmila Gorbachovaa, student, Faculty of Computer Science, V.N.Karazin Kharkiv National University, Kharkov, Ukraine.

E-mail: [lusyag23@gmail.com](mailto:lusyag23@gmail.com)

Katerina Kuznetsova, student, Faculty of Computer Science, V.N.Karazin Kharkiv National University, Kharkov, Ukraine.

E-mail: [kate7smith12@gmail.com](mailto:kate7smith12@gmail.com)

**Comparative analysis and study of the properties of information carriers for steganographic data hiding in clustered file systems.**

**Abstract.** The paper studies and analyzes various modern information storage technologies, namely HDD, Flash-USB, SSD. We've analyzed different indicators such as the number of implemented products, price, speed of reading and writing. Besides, we've considered some indicators of the information carriers' efficiency in terms of view of the possibility of using steganographic methods for hiding information in clustered file systems. It has been analyzed the speed of sequential reading / writing and the speed of access to a random cluster, corresponding to the speed of access to a fragmented file. For this task, we used the test results from the UserBenchmark resource. The testing has made using the Sequential and Random4k methods. In addition, we have provided an assessment of information carriers and have gave recommendations of using the particular information carrier and method for hiding data by mixing clusters in the structure of the file system. Besides, it was analyzed the dependence of the speed parameters of access to the cluster on the level of file fragmentation. Refinements are made of how an increase or decrease in the level of fragmentation (entanglement) affects the speed of access to the file, which is an important indicator when using the method of hiding data in the file system structure. The advantages and disadvantages of various types of storage devices have been discussed, and its comparative analysis was made. Moreover, we analyzed the features of the process of defragmentation of drives, and the influence of various factors on the overall level of fragmentation on the storage medium. We placed emphasis on the greater the level of fragmentation on the storage medium, the more information could be hidden. It was concluded that due to the widespread use of SSD / HDD drives, the method of hiding information in the structure of file systems, by mixing clusters of covering files, is relevant.

**Keywords:** File storage media; Fragmentation; Speed of access; Data hiding, Steganographic.

**Рецензент:** Николай Карпинский, д.т.н., проф., Университет Бельсько-Бяла, ул. Виллова 2, 43-309 Бельсько-Бяла, Польша.

E-mail: [mkarpinski@ath.bielsko.pl](mailto:mkarpinski@ath.bielsko.pl)

Поступила: Декабрь 2020.

**Авторы:**

Кирил Шеханин, аспирант, Харьковский национальный университет имени В. Н. Каразина, Харьков, Украина.

Людмила Горбачова, студентка факультета компьютерных наук, Харьковский национальный университет имени В. Н. Каразина, Харьков, Украина.

E-mail: [lusyag23@gmail.com](mailto:lusyag23@gmail.com)

Екатерина Кузнецова, студентка факультета компьютерных наук, Харьковский национальный университет имени В. Н. Каразина, Харьков, Украина.

E-mail: [kate7smith12@gmail.com](mailto:kate7smith12@gmail.com)

**Сравнительный анализ и изучение свойств носителей информации для стеганографических данных, скрываемых в кластерных файловых системах.**

**Аннотация:** В работе рассмотрены современные технологии хранения информации, а именно *HDD, Flash-USB, SSD*. Проанализированы такие показатели, как количество реализованных изделий, цена, скорость считывания и записи. Исследованы некоторые показатели эффективности носителей информации, с точки зрения возможности применения стеганографических методов сокрытия информации в кластерных файловых системах. Выполнен анализ скорости последовательного чтения / записи и скорости доступа к случайному кластеру, соответствующей скорости доступа к фрагментированному файлу. Для этого использовались результаты тестирования с ресурса *UserBenchmark*. Тестирование выполнялись методами *Sequential* и *Random4k*. Предложена оценка носителей информации и даны рекомендации по использованию носителя информации и метода сокрытия данных путем перемешивания кластеров в структуре файловой системы. Проведен анализ взаимосвязи и зависимости параметров скорости доступа к кластеру от уровня фрагментированности файла. Уточнено, каким образом увеличение или уменьшение уровня фрагментированности (*переплетенности*) влияет на скорость доступа к файлу, что является важным показателем при использовании метода сокрытия данных в структуре файловой системы. Рассмотрены преимущества и недостатки накопителей информации различных типов, и проведен их сравнительный анализ. Выполнен анализ особенностей процесса дефрагментации накопителей, и влияние различных факторов на общий уровень фрагментированности на носителе информации. Подчеркнуто, что чем больше уровень фрагментированности на носителе данных, тем больше информации можно скрыть. Утверждается, что благодаря широкому распространению *SSD / HDD* накопителей, метод сокрытия информации в структуру файловых систем, путем перемешивания кластеров покрывающих файлов, является актуальным.

**Ключевые слова:** файловые носители информации; фрагментация; скорость доступа; сокрытие данных; стеганография.