УДК 004.49
DOI 10.47049/2226-1893-2025-4-226-240

ВИКОРИСТАННЯ PDF ФАЙЛІВ В ЯКОСТІ СТЕГАНОГРАФІЧНИХ КОНТЕЙНЕРІВ

Ю.В. Даус
к.геогр.наук, доцент кафедри «Технічна кібернетика
й інформаційні технології ім. проф. Р.В. Мерката»
ORCID: 0000-0001-9737-4663

Одеський національний морський університет, Одеса, Україна

Анотація. *В наш час стеганографія використовує практично все різноманіття стеганографічних контейнерів починаючи від звичайних графічних файлів до відео та аудіо контенту.*

Задача стеганографії приховати сам факт передачі сповіщення чи якоїсь інформації. З іншого боку, потрібно бути впевненим що передана вами інформація не була спотворена чи заміщена.

Часто, щоб захистити від спотворення документу на шляху до кінцевого користувача використовують формат файлу pdf – portative document format. Однак, цей формат файл внаслідок доволі простої та зручної будови доволі легко використовувати в якості стеганографічного контейнера та дописувати значні обсяги інформації без відома відправника файлу. Але ми можемо використати ці властивості не тільки для передачі прихованої інформації, але й для захисту самої інформації передаючи разом з самим файлом хеш самого файлу.

Ключові слова: *стеганографія; вразливості; кібербезпека, стеганографічні контейнери, pdf стеганоконтейнер.*

UDC 004.32.26:004.056.523
DOI 10.47049/2226-1893-2025-4-226-240

USING PDF FILES AS STEGANOGRAPHIC CONTAINERS

Y.V. Daus
Ph.D., docent of the Department «Technical Cybernetics
and Information Technologies named after prof. R.V. Merkt»
ORCID: 0000-0001-9737-4663

Odesa National Maritime University, Odesa, Ukraine

Abstract: *Nowadays, steganography uses almost all the variety of steganographic containers, from ordinary graphic files to video and audio content. The task of steganography is to hide the very fact of transmitting a notification or some information. On the other hand, you need to be sure that the information you have transmitted has not been distorted or replaced. Often, to protect the document from distortion on the way to the end user, the pdf file format is used – portable document format. However, this file format, due to its rather simple and convenient structure, is quite easy to use as a steganographic container and add significant amounts of information without the knowledge of the sender of the file. But we can use these properties not only to transmit hidden information, but also to protect the information itself by transmitting the hash of the file itself along with the file itself.*

Keywords: *steganography; vulnerabilities; cybersecurity, steganographic containers, pdf steganocontainer.*

Вступ. У наш час кількість інформації що передається в інформаційних мережах постійно збільшується. Але з'являється все більше сервісів що відстежують та контролюють значні об'єми цієї інформації. Часто аналіз цієї інформації використовується в розвідувальних органах різних країн. Окремо, не залежно від державних органів, таку інформацію використовують в OSINT (Open Source Intelligence) – розвідка з відкритих джерел. В OSINT збирається багато інформації і з використанням штучного інтелекту можна зробити приголомшливі висновки на основі відкритих даних. Тому, під час дії військового стану, бажано приховувати деяку інформацію, на основі якої можна зробити висновки про стан економіки, про основні показники виробництва в промисловості, кількість виробленої та спожитої електроенергії, об'єм перевезення автотранспортом, залізницею тощо.

Деякі дані нам необхідно передавати, наприклад підрядникам для планування та підготовки необхідних ресурсів, сервісним та ремонтним службам. Щоб передавати такі дані відкритими каналами ми можемо використати стеганографічні контейнери, в які можемо вкладати необхідні дані та вже потім передавати відкритими каналами: електронною поштою, месенджерами, сповіщеннями.

Стеганосистема являє собою комплекс програмних, апаратних або програмно-апаратних засобів, що забезпечують приховування та вилучення даних на базі того чи іншого цифрового носія.

Відповідно, головним завданням будь-якої стеганосистеми незалежно від середовища, у якій вона функціонує та особливостей її реалізації є адаптація приховуваних даних до середовища, у яке їх буде поміщено, безпосередньо вбудовувано та з наступним вилученням [1; 2].

Стеганографія може працювати з різними видами файлів та медіаконтентом [2]. Основні типи файлів, з якими стеганографія може працювати, включають:

1. Зображення – це один з найпоширеніших видів носіїв даних. Приховану інформацію можна безпосередньо вбудувати в саме зображення або в метадані. В деяких форматах зображень можливо дописувати інформацію в кінець зображення. Змінюючи молодші пікселі в зображені ми можемо здійснити деяку корекцію

зображення яка не помітна людському оку. Зміна метаданих дуже помітна при перегляді властивостей файлу, а також містить доволі мало об'єму для приховування сповіщень.

2. Аудіо та відео – у цьому випадку прихована інформація може бути вбудована в аудіосигнал чи відеосигнал, змінюючи амплітуду чи частоту деяких частотних компонентів та вбудовування в відеокадри.

3. Текст – можна використовувати текстові файли, вставляючи приховану інформацію в текстовий контент, інтервали між словами або різні атрибути тексту, такі як розмір шрифту, колір тексту тощо.

4. Мережевий трафік – приховані дані можуть бути вбудовані в мережевий трафік, використовуючи певні властивості пакетів.

5. Інші формати – стеганографічним контейнером тут можуть виступати різні формати даних, включаючи PDF-файли (portable document format), ZIP-архіви, електронні документи, а також в інші креативні стеганоконтейнери, в залежності від конкретного завдання та потреби в приховуванні інформації.

З нашої точки зору, хочеться ще поділити стеганографічні контейнери на категорії за якістю по наступним параметрами:

- рівномірність приховуванням повідомлення;
- рідкість використання в якості стеганоконтейнера;
- ємність контейнера;
- важкість руйнування контейнера.

Доволі зручно проаналізувати всі ці оцінки у вигляді таблиці. Оцінювати будемо за п'ятибальною системою де 1 – самий низький бал а 5 – самий високий бал. Результати такої оцінки представлені в таблиці 1.

Таблиця 1

Оцінка якості стеганоконтейнера

№	Назва стегано-котейнера	Рівно-мірність	Рідкість викорис-тання	Ємність контей-нера	Важкість руйнування	Загальний бал
1	Зображення	5	2	4	1	12
2	Аудіо та відео	4	3	5	2	14
3	Текст	1	4	3	1	9
4	Мережевий трафік	3	5	5	5	18
5	PDF та ZIP файли	4	4	4	4	15

Найбільш можлива оцінка – 20 балів, а мінімальна – 4 бали. Найбільшу оцінку в 18 балів отримав мережевий трафік. Знижена оцінка рівномірності заповнювання контейнера оскільки ми можемо посилати додаткову інформацію тільки в окремих специфічних пакетах. Ми не можемо відсилати багато пакетів,

адже це може привернути увагу програм по відслідковування мережевого трафіка, особливо якщо трафік йде за межі внутрішньої мережі. Проте не всі організації встановлюють такі моніторингові програми та аналізують мережевий трафік.

Друге місце з оцінкою 15 балів отримали контейнери у вигляді PDF файлів та архівів. Рівномірність заповнення – 4 бали. Оцінка знижена в зв'язку з нерівномірністю заповнення файлів архіву. В PDF файлах можна практично рівномірно заповнити весь файл і навіть дописати додаткову інформацію в кінець файлу. Такі файли не часто використовують в якості стеганоконтейнера але вони дуже часто пересилаються електронною поштою та через месенджери. Ємність контейнера тут доволі висока 30-50 % від початкового розміру контейнера. Зазвичай, користувачі дуже рідко співвідносять розмір файлу та об'єм інформації в ньому. Розмір файлу важко оцінити, оскільки одні і ті самі об'єкти можуть додаватися як текст і також як текст у вигляді зображення. Ну і остання оцінка по важкості руйнування – при перезапису файлу може бути пошкоджена тільки та частина що дописується в кінець файлу. Частина що міститься всередині файлу сприймається як частина цього файлу і повинна бути перезаписана разом з іншими об'єктами.

Третє місце по якості з оцінкою 14 балів посідає стеганоконтейнер з аудіо та відео інформацією. Рівномірність заповнення 4 бали. Не даємо найвищу оцінку тому що, як правило, це припадає на окрему доріжку або зміну амплітуди звукового каналу. Ємність каналу дуже висока оскільки розмір таких файлів величезний. Але тут постає питання передачі таких великих файлів. Їх розмір одразу привертає до себе увагу і може викликати труднощі при пересиланні електронною поштою, оскільки зазвичай система встановлює обмеження на розмір файлу. Контейнер доволі легко руйнується при переписуванні файлів в іншій роздільній якості та при додаванні додаткової звуковою доріжки, наприклад додавання мови перекладу.

Четверте місце по якості займає стеганоконтейнер у вигляді зображення – 12 балів. Цей контейнер дуже часто використовується при передачі прихованих сповіщень і всі звертають на нього увагу. Часта передача картинок в сповіщеннях стає тим самим червоним маркером для кіберфахівців що привертає їх увагу. Доволі часто в системах безпеки інформаційних систем стоять фільтри, які автоматично при знаходженні в файлах електронної пошти зображень змінюють в них молодші біти. Така заміна руйнує стеганоконтейнер і не дозволяє передати приховане сповіщення, хоча саме зображення буде доставлене адресату.

І останнє місце по якості стеганоконтейнера з оцінкою 9 балів займають текстові файли. Найбільше оцінки знижені за рівномірність заповнення контейнера та важкість руйнування. В цих файлах приховане сповіщення ми можемо вписати в атрибути тексту, в метадані, колір тексту. Але все це легко руйнується при перезаписувані файлу стандартними інструментами. Об'єм контейнера залежить від розміру файлу, але не більше 10-15%. Додавання більшого об'єму даних призводить до великого збільшені розміру файлу та стає помітним навіть не фахівцям. Текстові файли та документи доволі часто пересилаються електронною поштою та різноманітними месенджерами і не привертають до себе великої уваги.

У нашій роботі ми хочемо обрати простий та надійний стеганоконтейнер, що дозволить не привертаючи увагу передавати великі об'єми прихованих сповіщень. Метод вбудовування прихованих сповіщень повинен бути дуже простим, не вимагати багато обчислювальних ресурсів, міг передаватися звичайними відкритими каналами та не привертати до цього контейнеру додаткової уваги. Ми також хочемо, щоб отримувач міг пересвідчитися в оригінальності отриманого сповіщення і відсутності підміни.

Вибір стеганоконтейнера. Розділимо нашу задачу на кілька етапів. Спочатку виберемо перспективний стеганографічний контейнер враховуючи кілька параметрів. Далі вивчимо структуру цього контейнера та можливі місця вбудовування прихованих сповіщень. Визначимося з методом шифрування сповіщень та передачею шифрувальних ключів. І на останок, визначимо метод перевірки файлів на підміну сповіщень. Все це реалізуємо в єдиному програмному комплексі.

За оцінками якості на перше місце виходить стеганоконтейнер з мережевим трафіком. Але великі мінуси такого контейнера це підбір підходящих пакетів для передачі та дозволи на передачу інформації за межі внутрішньої інформаційної комп'ютерної мережі. Також потрібна розробка доволі складного програмного забезпечення для відслідковування таких пакетів та витягування прихованих сповіщень.

Другим, за оцінками якості, є контейнери у вигляді PDF файлів та файли архівів. З переваг цих контейнерів:

- велика ємність;
- широко застосовуються при пересиланні електронною поштою та месенджерами;
- доволі проста структура файлу;
- стеганографічний контейнер важко зруйнувати стандартними інструментами.

Тому в нашій роботі ми будемо намагатися створити стеганографічний контейнер в PDF файлі вибираючи найбільш просте та очевидне рішення.

Структура PDF файлу. Спершу розглянемо спрощену структуру цього файлу та можливості вбудовування прихованого сповіщення.

Основна структура PDF

1. Заголовок (Header)

%PDF-1.7: Перший рядок файлу вказує версію PDF формату в даному випадку версія 1.7

2. Тіло (Body) може містити текст та об'єкти. Кожен об'єкт має унікальний номер та версію і містить всі дані документа.

3. Таблиця перехресних посилань (XRef Table):

- Містить offset кожного об'єкта у файлі;
- Дозволяє швидкий доступ до будь-якої частини PDF;
- Критично важлива для навігації.

4. Завершальний запис (Trailer):

- Містить посилання на кореневий об'єкт;

- Вказує позицію XRef таблиці;
- Список всіх сторінок документа;
- Kids – масив посилань на сторінки;
- Count – кількість сторінок;
- %% EOF – кінець файлу

Загалом знак % використовується в файлі як коментар або інформаційне повідомлення і сприймається як частина файлу. Подвійний знак проценту %% EOF означає кінець файлу. У версіях 1-1.7 інформація після знаку кінця файлу яка позначена коментарем (%) також сприймається.

На рисунку 1 приведений типовий заголовок файлу PDF. Перший рядок показує що ми використовуємо версію 1.3 (% PDF-1.3). Початок наступного рядка позначений символом % і сприймається як коментар. Далі йде тіло файлу і містить текст та різноманітні об’єкти. «10 obj» – це посилання на корневий каталог, «30 obj» – окремі сторінки, «40 obj» – потоки даних. Окремо в файлах PDF використовують дані наступних типів: boolean, number, string, array, dictionary, унікальний ідентифікатор та null – порожнє значення.

```
%PDF-1.3
%
3 0 obj
<< /Filter /FlateDecode /Length 821 >>
stream
x
endstream
endobj
1 0 obj
<< /Type /Page /Parent 2 0 R /Resources 4 0 R /Contents 3 0 R /MediaBox [0 0 595.
>>
endobj
4 0 obj
<< /ProcSet [ /PDF /Text ] /ColorSpace << /Cs1 5 0 R >> /Font << /TT2 7 0 R
/TT4 9 0 R >> >>
endobj
10 0 obj
<< /N 3 /Alternate /DeviceRGB /Length 2612 /Filter /FlateDecode >>
stream
```

Рис. 1. Типовий заголовок PDF файлу

Цікавим є кінець файлу, приклад якого наведено на рисунку 2. На рисунку видно таблицю посилань на об’єкти (xref), та трейлер файлу де вказується розмір та ідентифікатор. В кінці файлу стоїть подвійний знак проценту та директива EOF що означає кінець файлу.

```
@@M@@JFnI @ @9r@@a@5@>[nb @|@;@@B^bn*T@  ¦"@4@ @;&@@L @ 1 @@M} @^ @o@@C@ ,@@ @q6@@
&6@ @%@@@@a`c@ @( @@W@2@@T@@@@@ v@ AEAU @@@@@G@@@ V@@
, @@@@@A@@@@@@@ \@@2 s|CWu1@
endstream
endobj
18 0 obj
<< /Producer (macOS Version 26.0 \(Build 25A354\) Quartz PDFContext) /CreationDa
(D:20250919074300Z00'00') /ModDate (D:20250919074300Z00'00') >>
endobj
xref
0 19
0000000000 65535 f
0000000915 00000 n
0000003881 00000 n
0000000022 00000 n
0000001024 00000 n
0000003845 00000 n
0000000000 00000 n
0000004014 00000 n
0000000000 00000 n
0000021863 00000 n
0000001132 00000 n
0000003964 00000 n
0000004955 00000 n
0000004368 00000 n
0000005191 00000 n
0000022321 00000 n
0000022025 00000 n
0000022569 00000 n
0000022922 00000 n
trailer
<< /Size 19 /Root 11 0 R /Info 18 0 R /ID [ <596779c1aae7f028d6e7e3070500be42>
<596779c1aae7f028d6e7e3070500be42> ] >>
startxref
23085
%%EOF
```

Рис. 2. Типовий кінець PDF файлу

Зараз проведемо огляд методів застосування PDF файлів як стеганографічних контейнерів. В роботі Klemm & Chen [3] розглядається модифікація метода зміни найменш значущих бітів (LBS) у стрімах PDF файлу. При цьому використовують 32 оператора що приймають значення. Ці значення можна змінити

в десяткових числах оператора. Наприклад зміна числа 98.0 на 98.1 не може спричинити таких візуальних ефектів як зміна числа 98.0 на 99.0. При цьому слід враховувати версії PDF файлів, оскільки версії 1.3 - 1.7 використовують трохи іншу нумерацію операторів ніж версія 2.0

В роботі Ndounda R. & Ekodeck [4] використовують китайську теорему лишків (CRT) шляхом маніпуляцій символами та структурою PDF без явного порушення формату. Попередньо відправник та отримувач повинні домовитись про довжину блоку, на яку секретне повідомлення буде розділене перед кодуванням. Така інформація повинна передаватися через захищений канал щоразу перед відправленням PDF файлу з прихованим повідомленням. Це викликає багато незручностей для застосування даного методу.

Ще один метод розглядається в роботі Koptyra P.& Ogiela L [5], тут проходить модифікація структури самого файлу. Основна ідея полягає в тому, щоб видалити посилання на певну сторінку з її батьківської та зменшити лічильник сторінок. Вміст об'єкта залишається недоторканим, а видаляється лише посилання. Також автори розглядають можливість поділення секретного повідомлення на кілька, та розміщення в окремих файлах та збірка цього сповіщення з окремих файлів. Недоліком такого підходу, на нашу думку, є можливість втрати одного з файлів, що приведе до неможливості відтворення цього повідомлення.

В роботі [6] демонструється приховування виконавчих файлів зловмисного коду і обходу захисних механізмів PDF. В сучасних версіях PDF 2.0 цей механізм заблоковано з метою безпеки, але в нижчих версіях це все ще можливо.

Ще одним з методів вписати приховане повідомлення, є модифікація відступів та пробілів в файлі PDF методом зміни молодших бітів у об’єкті [7]. При вписуванні прихованого сповіщення модифікуються відступи, зміни практично зовсім не помітні для людського ока. Наприклад, зміну відступу початку абзацу з 1.25 на 1.252, або зміна відступу кожного рядку з 0.0 на 0.1 практично не можна зафіксувати без вимірювань. Недоліком такого підходу може бути перезаписування файлу та руйнування контейнера.

Вчені Maiorca D., Biggio B. в роботі [8] розглядаються можливість вписування шкідливого коду через вбудовані модулі java script. Це можна робити у версіях 1.0-1.7. Також таку вразливість можна використовувати для формування стеганографічного контейнеру в межах цього скрипту.

Загалом, PDF файли являють собою доволі гарний контейнер для прихованих сповіщень. Тому, розглянувши всі описані вище методи, вирішили дещо спростити процес вбудовування прихованих сповіщень в PDF файл. Наш метод повинен працювати дуже швидко, при перезапису контейнер не повинен руйнуватися, методи шифрування повинні враховувати передачу ключа шифрування прихованими каналами і через відкриті канали, також необхідно підтвердження оригінальності сповіщення.

Найпростішим способом вписування прихованого сповіщення у PDF файл, на нашу думку, є маскування самого сповіщення у вигляді коментаря. На жаль ми не знайшли робіт в яких би використовувалось маскування прихованого

сповіщення в коментарі PDF файлу. При використанні коментаря ми можемо поміщати приховане сповіщення практично в будь яке місце поточного файлу.

В структурі файлу PDF коментар у файлі починається з ключового символу % і закінчується знаком переходу на новий рядок. Кінець файлу позначають подвійним знаком процента %% EOF. Вважаємо, що гарним місцем для вписування прихованого сповіщення в коментар є місце перед ознакою кінця файлу. При перезаписуванні цього файлу через програмні оболонки різних переглядачів PDF файлів і подальшого його перегляду через спеціальні переглядачі типу «Atom» , «NotePad+» ми бачимо, що коментар зберігається у незмінному вигляді. Ми тестували версії PDF файлів від 1.0 до 1.7. Такий простий підхід має звичайно свої плюси та мінуси. Плюси що досить легко реалізувати на практиці, стеганоконтейнер досить ємнісний. З мінусів – можна доволі швидко знайти не відомий коментар при перегляді файлів спеціальними переглядачами. Якщо ви не створили цей файл, то не можете бути впевнені що цей фрагмент вписаний для приховування сповіщення. Але використовуючи надійне шифрування ми можемо не боятися виявлення сповіщення, оскільки без розшифрування практично не можливо дізнатися зміст самого сповіщення.

Проведений експеримент з вписуванням прихованого сповіщення після кінця файлу і з символом коментаря, також дав непогані результати. Сповіщення можна легко вилучити та прочитати. Але на жаль при перезаписуванні цього файлу до нового все що знаходилося після ознаки кінця файлу не записувалось (як і очікувалось) до нового файлу. Таку властивість ми можемо використати при оцінці факту підміни оригінального прихованого сповіщення.

Для збереження приватності та секретності, нам необхідно шифрувати наші сповіщення перед перенесенням до контейнеру. Доволі гарним рішенням є застосування симетричного методу шифрування AES з додаванням модифікатора та паролю. Це доволі стійкий та надійний метод симетричного шифрування. Ми розуміємо, що зараз практично можна розшифрувати все що завгодно, але затрачені зусилля, кошти та час не завжди будуть відповідати отриманому результату. В нашому випадку ми передаємо не розвіддані і не надто секретну інформацію. Тому втрата цих даних не буде для нас катастрофічною. Мінуси метода симетричного шифрування – необхідність надійної передачі ключа шифрування. Якщо є можливість передачі ключа шифрування альтернативними надійними каналами, то цей метод дуже підходить.

Ми також розглянули варіант, коли у нас не має надійного каналу передачі секретного паролю. В даному випадку ми пропонуємо застосувати альтернативний метод шифрування – асиметричний. Для асиметричних методів на сьогодні найбільш поширеними є метод RSA та метод шифрування по еліптичним кривим. При застосуванні цих методів створюються два ключі: публічний та приватний. Доступ до публічного ключа може бути необмежений. Як правило, цей ключ розміщають на якомусь сайті чи відкритому ресурсі. А приватний ключ відомий тільки власнику і зберігається тільки у нього. Таким чином, ми можемо зашифрувати повідомлення відкритим ключем отримувача, а отримувач зможе

розшифрувати його своїм приватним ключем. При застосуванні даного метода у нас зникає проблема надійної передачі ключів шифрування.

Для реалізації цього алгоритму використання PDF файлу ми розробили програмний комплекс «PdfMessage» рисунок 3.

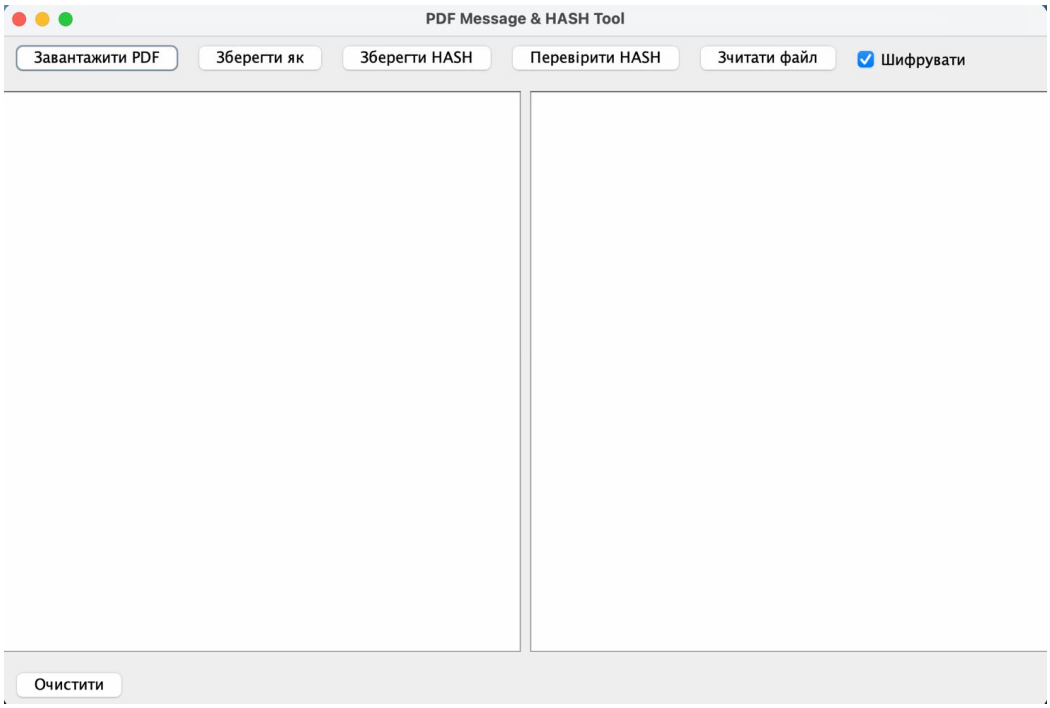

Рис. 3. Основне меню програмного комплексу PdfMessage

Для прикладу ми можемо вписати повідомлення в ліве віконце програми та використовуючи розділ «Зберегти як» записати модифікований файл PDF з нашою інформацією. При цьому користувач повинен вибрати тип шифрування в спливаючому вікні. Далі йде зберігання сповіщення в зашифрованому вигляді. На рисунку 4 показаний приклад роботи програми. В лівому вікні програми наше сповіщення, а в правому сповіщення що ми витягнули з файлу та розшифрували використовуючи розділ меню «Зчитати файл».

Звичайно, дуже важливою задачею є слідкування за можливістю підміни сповіщення та модифікації файлу з прихованим сповіщенням. Для цього ми скористаємося властивістю хеш функцій, яка говорить про те що при зміні навіть біта інформації в файлі, його хеш рядок кардинально змінюється. Звичайно можуть теоретично бути так звані зіткнення – коли два різні файли мають один і той же хеш рядок. Але вірогідність такого співпадіння настільки маленька, що ми не будемо це брати до уваги.

Алгоритм дій доволі простий. Після вписування прихованого сповіщення ми вираховуємо значення такої функції для цього файлу, включно зі знаком кінця файлу %%EOF. Але значення хеш рядка зашифруємо та запишемо вже після знаку кінця файлу задаючи перед ним символ %. Це ніяк не вплине ні на якість файлу ні особливо на розмір файлу. Для більшої зручності ми можемо придумати якусь комбінацію знаків з знаком початку коментаря %. Можливо це може бути четверний знак проценту %%%%, або якась комбінація, наприклад %###@@@. Нам потрібно підібрати таку комбінацію символів, яка б дуже рідко зустрічалась в файлах.

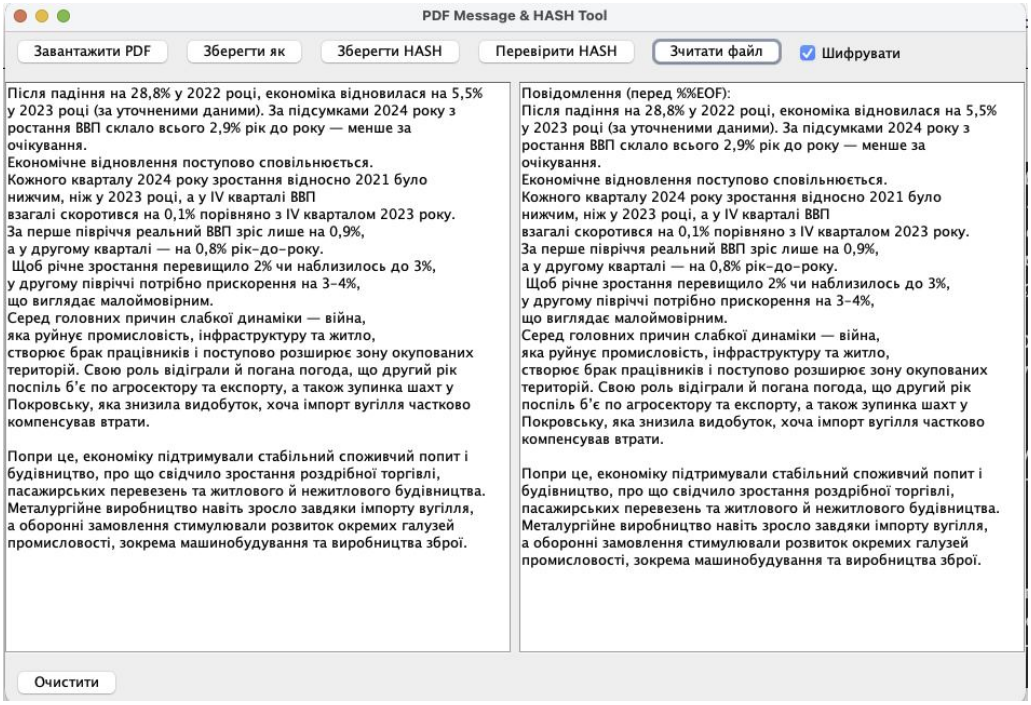

Рис. 4. Приклад зчитування прихованого сповіщення з файлу

Ми перевіряли вплив такого дописування на різноманітних переглядачах PDF файлів і всі працювали без збоїв. Версії файлів були від 1.3 до 1.7. Але при перезаписуванні файлу цей контент зникав, оскільки більшість інструментів припиняють зчитувати файл після досягнення символу кінця файлу. При зміні інформації всередині контейнера хеш рядок не зійдеться і ми будемо знати про підміну оригінального сповіщення. На рисунку 5 показано приклад перевірки хеш рядку файлу з вбудованим прихованим сповіщенням. В даному випадку наш обчислений хеш рядок та хеш рядок який був дописаний у файл з прихованим повідомлення співпадають і ми даємо відповідне сповіщення на зеленому фоні. При відсутності співпадіння це сповіщення буде на червоному фоні. При бажанні можна додати сюди також звуковий сигнал.

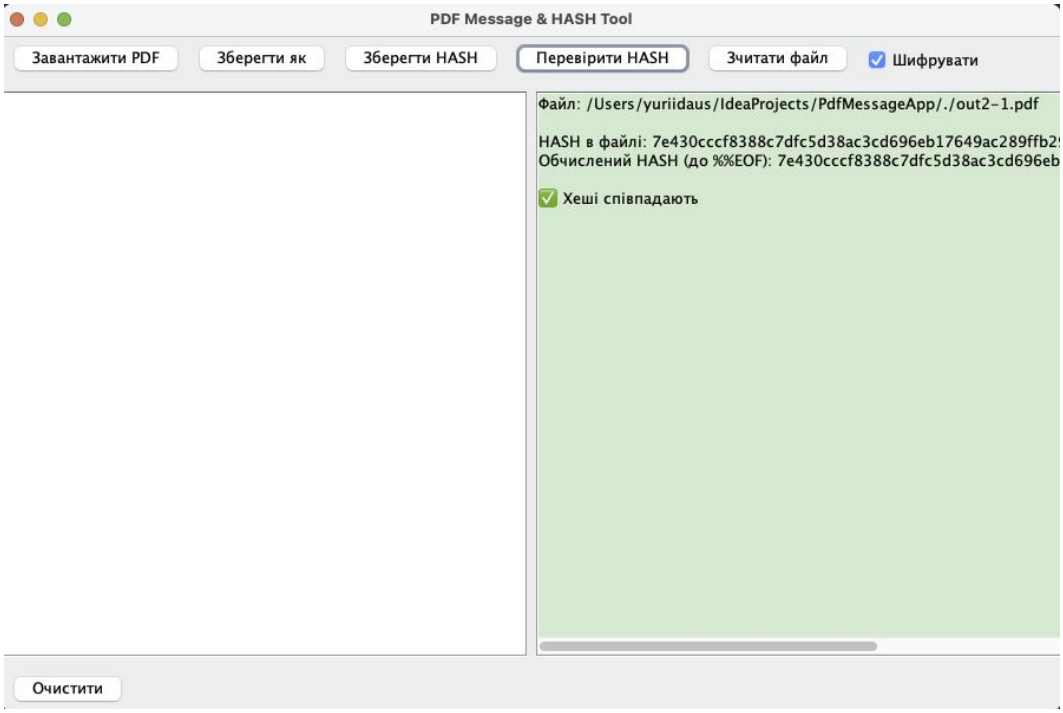

Рис. 5. Приклад порівняння поточного та записаного в файлі хеш рядка

Таким чином ми можемо контролювати модифікування файлів через які ми передаємо приховані сповіщення.

Висновки. Ми розглянули декілька стеганографічних контейнерів. Найбільш перспективними, на нашу думку, є мережевий трафік та PDF і ZIP файли. Різноманіття версій PDF файлів робить цей контейнер найбільш перспективним не тільки для вбудовування прихованих сповіщень, але й для вбудовування в цей файл різноманітного шпигунського програмного забезпечення та можливість його запуску при перегляді цих файлів.

Самим простим способом вбудовування прихованих сповіщень в PDF файл є оформлення таких сповіщень як коментар всередині PDF файлу. Для чіткого відокремлення від звичайних коментарів можна додавати якийсь специфічний символ після знаку %. Можливо це буде потрійний символ проценту або кілька специфічних символів підряд.

Додатково до вбудовування прихованих сповіщень, ми розглянули ознаки модифікації сповіщення через використання хеш функцій. Для цього ми після закінчення файлу підраховуємо хеш-рядок разом з вбудованим сповіщенням, шифруємо та дописуємо після кінця файлу. При перезаписувані файлу, його модифікації, зміненню самого прихованого сповіщення, цей хеш рядок або буде взагалі відсутній або хеш рядки не будуть відповідати поточному та переданому разом з прихованим сповіщенням. При перевірці хеш рядка та отримання не

відповідності довіряти такому сповіщенню не рекомендується. В такому випадку рекомендується проаналізувати та виявити на якому етапі виникла підміна та модифікація файлу. Створений нами застосунок доволі простий в експлуатації, дозволяє дуже швидко шифрувати та вбудовувати приховані сповіщення в PDF файли, контролювати підміну файлів та оригінальність сповіщення. Цей застосунок може бути корисним при передачі службової інформації між підрозділами одного підприємства, при пересиланні інформації підрядникам, пересиланні інформації між державними та приватними підприємствами та пересилання конференційних повідомлень між приватними особами.

ЛІТЕРАТУРА

1. Кондратюк, В.В., & Маркова, І.С. (2017). Стеганографія у сучасному інформаційному просторі // *Вісник Національного технічного університету України «Київський політехнічний інститут».* Серія: Радіотехніка, радіоапаратобудування, 1(71), 37-43.
2. Ковальчук, І.В., & Бондар, С.А. (2016). Стеганографічні методи захисту інформації у цифрових зображеннях. Молодь і ринок, 4 (142), 57-61.
3. Bryan, W.L., & Harter, N. (1897). Studies in the physiology and psychology of the telegraphic language. Psychological Review, 4(1), P. 27-53. https://doi.org/10.1037/h0073806.
4. Klemm R., Chen B. Hiding Sensitive Information Using PDF Steganography. arXiv preprint arXiv:2405.00865, 2024. Документ доступний: https://arxiv.org/pdf/2405.00865 та https://arxiv.org/html/2405.00865.
5. Ndoundam R., Ekodeck S.G.R. PDF Steganography based on Chinese Remainder Theorem. arXiv preprint arXiv:1506.01256, 2015. PDF доступний: https://arxiv.org/pdf/1506.01256 arXiv.
6. Koptyra P., Ogiela L. Distributed Steganography in PDF Files – Secrets Hidden in Modified Pages. PMC (PubMed Central). 2021. Доступ: https://pmc.ncbi.nlm.nih.gov/articles/PMC7517136/ PMC+2ResearchGate+2.
7. Vulnerability Exploitations Using Steganography in PDF Files. International Journal of Computer Networks and Applications (IJCNA). Доступний PDF: https://www.ijcna.org/Manuscripts/IJCNA-2020-O-02.pdf ijcna.org.
8. A new method for pdf steganography in justified texts. ScienceDirect. Доступ: https://www.sciencedirect.com/science/article/abs/pii/S2214212618 30485X ScienceDirect.
9. Maiorca D., Biggio B. Digital Investigation of PDF Files: Unveiling Traces of Embedded Malware. arXiv preprint arXiv:1707.05102, 2017. Доступ: https://arxiv.org/pdf/1707.05102 arXiv.
10. Steganography: An Overview. Geethanjali College of Engineering & Technology. PDF доступний на ResearchGate. ResearchGate.

11. A Comprehensive Review on Advancements and Applications of Steganography. ResearchGate. Доступ: https://www.researchgate.net/publication/379021602_A_Comprehensive_Review_on_Advancements_and_Applications_ of_Steganography ResearchGate.
12. Steganography A Data Hiding Technique. St. Cloud State University Institutional Repository. PDF: https://repository.stcloudstate.edu/context/msia_etds/article/1107/viewcontent/auto_convert.pdfrepository.stcloudstate.edu.

REFERENCES

1. Kondratyuk, V. V., & Markova, I. S. (2017). Steganography in the modern information space. Bulletin of the National Technical University of Ukraine «Kyiv Polytechnic Institute». Series: Radio Engineering, Radio Equipment Manufacturing, 1(71), 37-43.
2. Kovalchuk, I. V., & Bondar, S. A. (2016). Steganographic methods of information protection in digital images. Youth and Market, 4(142), P. 57-61.Chalaya L.E. User identification model based on keyboard handwriting. «Artificial Intelligence», No. 4. 2004, pp. 811-817.
3. Bryan, W. L., & Harter, N. (1897). Studies in the physiology and psychology of the telegraphic language. Psychological Review, 4(1), P. 27-53. https://doi.org/10.1037/h0073806
4. Klemm R., Chen B. Hiding Sensitive Information Using PDF Steganography. arXiv preprint arXiv:2405.00865, 2024. https://arxiv.org/pdf/2405.00865 та https://arxiv.org/html/2405.00865
5. Ndoundam R., Ekodeck S.G.R. PDF Steganography based on Chinese Remainder Theorem. arXiv preprint arXiv:1506.01256, 2015. https://arxiv.org/pdf/1506.01256 arXiv.
6. Koptyra P., Ogiela L. Distributed Steganography in PDF Files – Secrets Hidden in Modified Pages. PMC (PubMed Central). 2021.: https://pmc.ncbi.nlm.nih.gov/articles/PMC7517136/ PMC+2ResearchGate+2.
7. Vulnerability Exploitations Using Steganography in PDF Files. Interna-tional Journal of Computer Networks and Applications (IJCNA). https://www.ijcna.org/Manuscripts/IJCNA-2020-O-02.pdf ijcna.org.
8. A new method for pdf steganography in justified texts. ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S221421261830485X ScienceDirect.
9. Maiorca D., Biggio B. Digital Investigation of PDF Files: Unveiling Traces of Embedded Malware. arXiv preprint arXiv:1707.05102, 2017. https://arxiv.org/pdf/1707.05102 arXiv.
10. Steganography: An Overview. Geethanjali College of Engineering & Technology. ResearchGate. ResearchGate.

11. A Comprehensive Review on Advancements and Applications of Steganography.ResearchGate. https://www.researchgate.net/publication/3790-21602_A_Comprehensive_Review_on_Advancements_and_Applications_of_Steganography ResearchGate.
12. Steganography A Data Hiding Technique. St. Cloud State University Institutional Repository. https://repository.stcloudstate.edu/context/ msia_etds/article/1107/ viewcontent/ auto_convert. PDFrepository. stcloudstate.edu.

Стаття надійшла до редакції 05.11.2025

Посилання на статтю: Даус Ю.В. Використання PDF файлів в якості стеганографічних контейнерів // *Вісник Одеського національного морського університету:* Зб. наук. праць, 2025. № 4 (78). С. 226-240. DOI 10.47049/2226-1893-2025-4-226-240.

Article received 05.11.2025

Reference a Journal Artic: Daus Y.V. Using PDF files as steganographic containers // *Herald of the Odesa National Maritime University*: Coll. scient. works, 2025. № 4 (78). P. 226-240. DOI 10.47049/2226-1893-2025-4-226-240.