

Ukrywanie danych tekstowych w pliku muzycznym bez zauważalnej dla człowieka utraty jakości odtwarzanego dźwięku.

## Spis treści:

<b>1. Wstęp.....</b>	<b>3</b>
<b>2. Omówienie formatów dźwięku .....</b>	<b>3</b>
2.1 PCM.....	3
2.2 WAV.....	3
2.3 AIFF .....	3
<b>3. Metoda LSB.....</b>	<b>3</b>
<b>4. Metoda Parity Coding .....</b>	<b>4</b>
<b>5. Działanie aplikacji.....</b>	<b>4</b>
<b>6. Porównanie .....</b>	<b>5</b>
6.1 Muzyka poważna .....	5
6.2 Muzyka rockowa i metalowa.....	6
6.3 Muzyka akustyczna z jednym instrumentem.....	6
6.4 Różne formaty plików .....	6
<b>7. Podsumowanie .....</b>	<b>7</b>
<b>8. Bibliografia.....</b>	<b>7</b>

## 1. Wstęp

Steganografia jest zaawansowaną techniką ukrywania wiadomości w plikach w taki sposób, aby nie były one wykrywalne przez nieuprawnione osoby. W odróżnieniu od kryptografii, która koncentruje się na szyfrowaniu treści wiadomości, steganografia ukrywa sam fakt istnienia informacji. Dzięki temu możliwe jest bezpieczne przekazywanie danych, które są praktycznie niewidoczne dla osób trzecich. W programie skupimy się na ukrywaniu danych tekstowych w plikach dźwiękowych. W szczególności omówimy formaty dźwięku takie jak PCM, WAV i AIFF, które są powszechnie używane w przemyśle audio. Podczas szyfrowania danych zastosowane zostaną metody najmniej znaczącego bitu (LSB) oraz ukrywanie wiadomości w echu (echo hiding). [1]

## 2. Omówienie formatów dźwięku

### 2.1 PCM

PCM (Pulse Code Modulation) jest formatem cyfrowego kodowania dźwięku, który polega na próbkowaniu analogowego sygnału dźwiękowego w regularnych odstępach czasu i przekształceniu go na dyskretne wartości. PCM nie stosuje kompresji stratnej, co zapewnia wysoką jakość dźwięku, ponieważ nie ma utraty informacji. Pliki PCM są zazwyczaj duże, ponieważ przechowują surowe dane dźwiękowe.

### 2.2 WAV

WAV (Waveform Audio File Format) jest formatem dźwięku, który może przechowywać dane dźwiękowe skompresowane lub nieskompresowane, najczęściej w formacie PCM. Jest szeroko wspierany przez systemy operacyjne, oprogramowanie do edycji dźwięku oraz odtwarzacze multimedialne. WAV zapewnia bardzo wysoką jakość dźwięku.

### 2.3 AIFF

AIFF (Audio Interchange File Format) jest formatem dźwięku opracowanym przez Apple, szeroko stosowanym w systemach macOS oraz w oprogramowaniu do edycji dźwięku na platformie Apple. AIFF najczęściej przechowuje nieskompresowane dane PCM, co zapewnia wysoką jakość dźwięku. Format ten może również przechowywać skompresowane dane audio,

## 3. Metoda LSB

Metoda LSB wykorzystuje fakt, że zmiana najmniej znaczących bitów w danych binarnych, takich jak obraz lub dźwięk, w niewielkim stopniu wpływa na ogólną jakość tych danych, a zmiany te są praktycznie niedostrzegalne dla ludzkiego oka lub ucha. W przypadku plików dźwiękowych, sygnał dźwiękowy jest reprezentowany przez próbki, z których każda jest kodowana w formie binarnej. [2]

### Ukrywanie informacji:

- Metoda LSB polega na zastępowaniu najmniej znaczących bitów (najmłodszych bitów) danych obrazu lub dźwięku bitami informacji, którą chcemy ukryć.
- Na przykład, jeśli chcemy ukryć tekst w obrazie, każdy znak tekstu jest zamieniany na jego odpowiednik binarny. Następnie bity tego znaku są wstawiane do najmłodszych bitów kolejnych pikseli obrazu.

## 4. Metoda Parity Coding

Kodowanie parzystości, jest techniką używaną głównie do wykrywania i korekcji błędów w transmisji danych. Metoda ta wykorzystuje dodawanie dodatkowych bitów, zwanych bitami parzystości, które pomagają określić, czy liczba jedynek w określonym zestawie bitów jest parzysta czy nieparzysta. Kodowanie parzystości jest stosowane zarówno w przypadku pojedynczych bitów (parzystość bitu) jak i całych bloków danych (parzystość blokowa).[3]

### Bit parzystości:

- Bit parzystości jest dodatkowym bitem dodawanym do bloku danych, którego zadaniem jest upewnienie się, że liczba jedynek w bloku danych jest parzysta (dla parzystości parzystej) lub nieparzysta (dla parzystości nieparzystej).
- **parzysta:** Bit parzystości jest ustawiany tak, aby całkowita liczba jedynek była parzysta.
- **nieparzysta:** Bit parzystości jest ustawiany tak, aby całkowita liczba jedynek była nieparzysta.

## 5. Działanie aplikacji

Aplikacja do steganografii audio pozwala na ukrywanie i odkrywanie tajnych wiadomości w plikach audio, takich jak WAV, AIFF, oraz PCM. W aplikacji zastosowano dwie techniki steganograficzne: LSB (Least Significant Bit) oraz kodowanie parzystości (Parity Coding). Aplikacja została napisana w języku Python, wykorzystując biblioteki takie jak wave, aifc, numpy, scipy, tkinter oraz tkinterdnd2 do tworzenia interfejsu graficznego i manipulacji plikami audio.

Działanie aplikacji:

### 1. Interfejs graficzny:

- Aplikacja posiada intuicyjny interfejs graficzny, który umożliwia użytkownikowi wybór pliku audio, wpisanie wiadomości lub wybranie pliku tekstowego z wiadomością oraz wybór metody kodowania.
- Użytkownik może przeciągnąć i upuścić plik audio do odpowiedniego pola, przeglądać pliki na swoim komputerze, aby wybrać plik audio lub tekstowy.

### 2. Kodowanie wiadomości:

- Użytkownik wybiera plik audio, wpisuje wiadomość lub wybiera plik tekstowy z wiadomością, a następnie wybiera metodę kodowania (LSB lub kodowanie parzystości).
- Po naciśnięciu przycisku "Zakoduj", aplikacja przetwarza plik audio i zapisuje zmodyfikowany plik z ukrytą wiadomością.

### 3. Dekodowanie wiadomości:

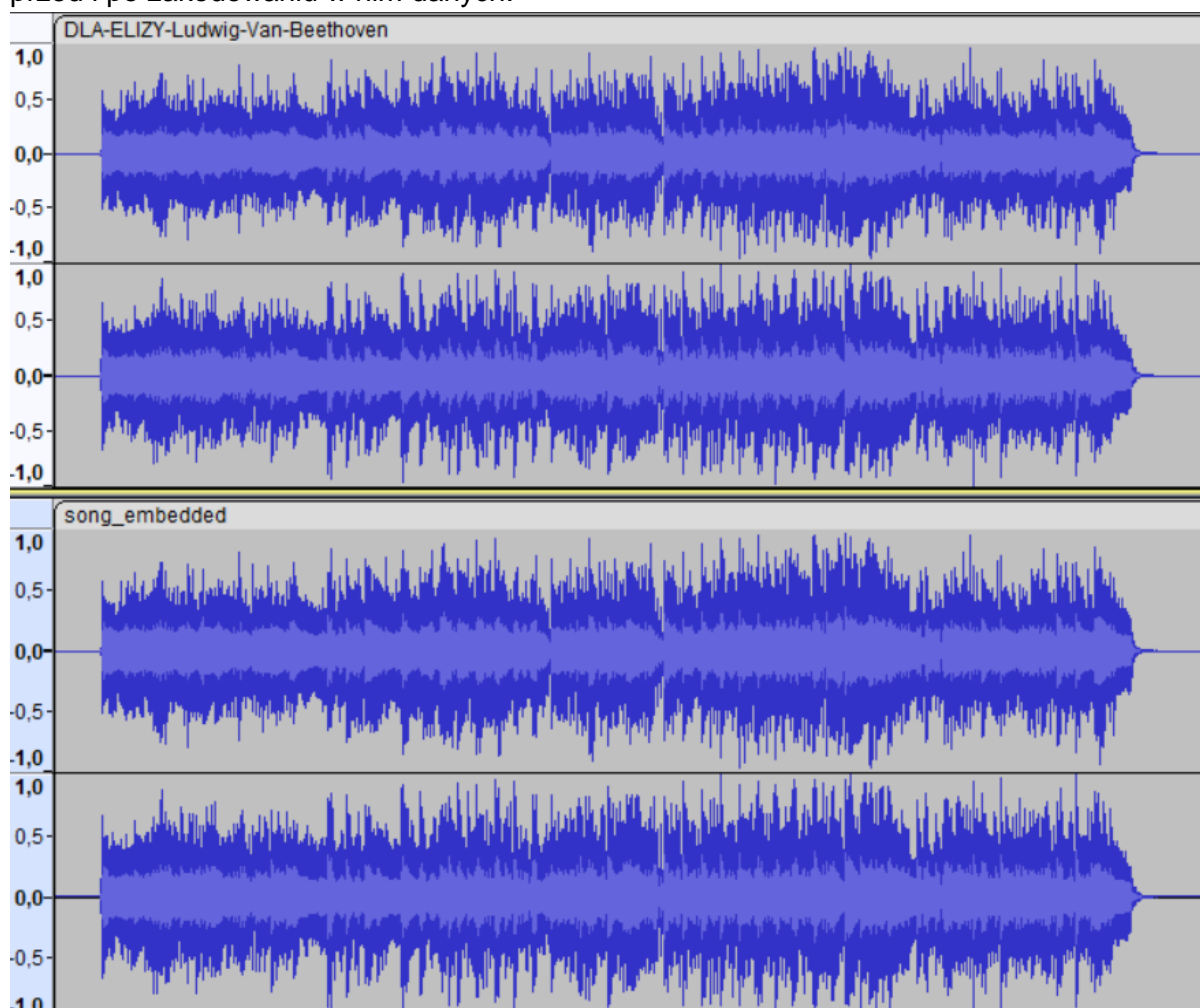
- Użytkownik wybiera plik audio z ukrytą wiadomością oraz metodę dekodowania.
- Po naciśnięciu przycisku "Odkoduj", aplikacja odczytuje ukrytą wiadomość i wyświetla ją w oknie dialogowym.

## 6. Porównanie

Podczas testowania użytych zostało kilka rodzaj muzyki. Poniżej przedstawione są wyniki dla kilku z nich.

## 6.1 Muzyka poważna

Jednym z nich była muzyka poważna, użyty utwór to Dla Elizy autorstwa Ludwiga van Beethovena. Podczas porównywania utworu przed zakodowaniem w nim informacji i po zakodowaniu, w metodzie LSB szum był wyraźnie słyszalny natomiast w Parity Coding był on minimalnie słyszalny. Poniżej przedstawiony jest wykres fali dźwiękowej dla utworu Dla Elizy przed i po zakodowaniu w nim danych:



*Rysunek 1 Wykres fali dźwiękowej*

Aby ocenić wpływ zakodowania danych na jakość dźwięku, przeprowadzona została szczegółowa analiza wykresów fal dźwiękowych przed i po procesie zakodowania. Na pierwszy rzut oka, przy standardowym poziomie powiększenia, nie było widać zauważalnych różnic między oryginalnym a zmodyfikowanym plikiem dźwiękowym. Wykresy fal dźwiękowych wyglądały niemal identycznie, co sugeruje, że metoda ukrywania danych nie wprowadza istotnych zmian, które byłyby zauważalne dla ludzkiego ucha w normalnych warunkach odsłuchu. Jednakże, po dokładniejszym przybliżeniu wykresu do poziomu dziesięciotysięcznych części sekundy, zaczęły być widoczne drobne różnice w amplitudzie fal dźwiękowych. Te subtelne zmiany są wynikiem modyfikacji poszczególnych bitów danych dźwiękowych w procesie zakodowania informacji. Przykładowe różnice zaznaczone są na rysunku 2.



*Rysunek 2 Różnice w wykresach*

## 6.2 Muzyka rockowa i metalowa

W celu dalszej weryfikacji wpływu procesu zakodowania danych na jakość dźwięku, przeprowadzono testy na wybranych utworach rockowych i metalowych. Utwory te charakteryzują się złożonymi aranżacjami, dynamicznymi partiami instrumentów i intensywnymi wokalami, co czyni je idealnymi kandydatami do oceny skuteczności metody ukrywania danych. Podczas testów, zarówno przed jak i po zakodowaniu danych, odsłuchiowano utwory rockowe, nie wykryto żadnych słyszalnych różnic w jakości dźwięku. Zarówno w przypadku cichych, jak i głośniejszych partii, zmodyfikowane pliki audio zachowały swoją oryginalną jakość, nie wykazując żadnych zauważalnych artefaktów czy zniekształceń.

## 6.3 Muzyka akustyczna z jednym instrumentem

W ramach oceny wpływu zakodowania danych na jakość dźwięku, przeprowadzono testy na utworach akustycznych z jednym instrumentem. Utwory te są szczególnie wrażliwe na wszelkie zakłócenia, ponieważ charakteryzują się czystym i nieskomplikowanym dźwiękiem, co ułatwia wykrycie nawet najmniejszych zmian. W wyniku testów stwierdzono, że po zakodowaniu danych w utworach akustycznych pojawił się słyszalny szum, który nie był obecny w oryginalnych nagraniach. Szum ten był najbardziej zauważalny w cichych partiach utworów, gdzie brak innych dźwięków uwypuklał zmiany wprowadzone przez proces ukrywania danych.

## 6.4 Różne formaty plików

Podczas przeprowadzonych testów oceniano wpływ formatu dźwięku na jakość audio po zakodowaniu danych. W badaniach wykorzystano trzy formaty: PCM, WAV oraz AIFF. Każdy z tych formatów jest szeroko stosowany w przemyśle audio i charakteryzuje się wysoką jakością dźwięku, ponieważ przechowują nieskompresowane dane audio. Wyniki testów wykazały, że wybór formatu dźwięku nie miał znaczącego wpływu na słyszalność szumu po zakodowaniu danych. Niezależnie od tego, czy używany był format PCM, WAV czy AIFF, jakość dźwięku po zakodowaniu danych pozostawała na porównywalnym poziomie. Szum wprowadzony przez proces ukrywania danych był równie słyszalny we wszystkich testowanych formatach, szczególnie w utworach akustycznych z jednym instrumentem.

## 7. Podsumowanie

Projekt koncentrował się na ukrywaniu danych tekstowych w plikach dźwiękowych bez zauważalnej utraty jakości odtwarzanego dźwięku. Omówiliśmy formaty dźwięku (PCM, WAV, AIFF) oraz metody steganograficzne: najmniej znaczącego bitu (LSB) i kodowanie parzystości (Parity Coding). Testy wykazały, że metoda LSB powoduje wyraźny szum w muzyce poważnej, natomiast Parity Coding generuje mniej zauważalne zakłócenia. W utworach rockowych i metalowych nie wykryto słyszalnych różnic, ale w muzyce akustycznej z jednym instrumentem szum był najbardziej wyraźny. Wybór formatu (PCM, WAV, AIFF) nie miał znaczącego wpływu na jakość dźwięku po zakodowaniu danych. Podsumowując, ukrywanie danych w plikach dźwiękowych jest efektywne, ale wpływ na jakość dźwięku zależy od rodzaju muzyki i metody steganograficznej.

## 8. Bibliografia

- [1]<https://en.wikipedia.org/wiki/Steganography>, Data dostępu: 17.06.2024
- [2]<https://sumit-arora.medium.com/audio-steganography-the-art-of-hiding-secrets-within-earshot-part-2-of-2-c76b1be719b3>, Data dostępu: 17.06.2024
- [3][https://en.wikipedia.org/wiki/Parity\\_bit](https://en.wikipedia.org/wiki/Parity_bit), Data dostępu: 17.06.2024