

Podstawy steganografii i steganoanalizy

Dominik Lau, Sebastian Kutny, Tomasz Lewandowski, Maciej Krzyżanowski

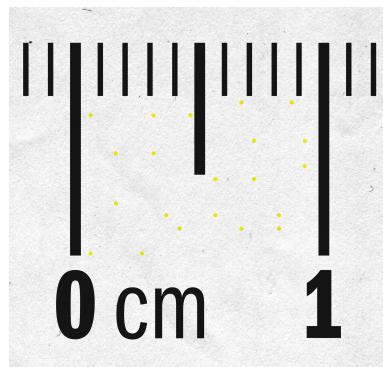
7 maja 2023

1 Czym jest steganografia? Do czego służy?

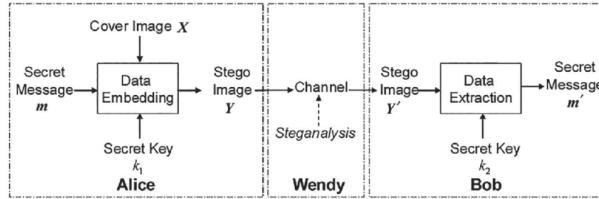
Steganografia polega na ukrywaniu informacji przez ukrywanie komunikacji w innej formie transmisji danych np. w obrazkach, plikach dźwiękowych, tekstowych. Zastosowania steganografii

- omijanie cenzury/szpiegostwo
- umieszczanie znaków wodnych
- ukryta wymiana danych
- dodawanie metadanych do plików (np. znaki sterujące)
- numery seryjne drukarek (za pomocą małych kropek)
- wprowadzanie opóźnień w pakietach sieciowych
- zastosowania w VoIP (steganofonia)
- zabezpieczanie banknotów (np. EURion constellation)

Steganografia może zatem realizować następujące funkcje bezpieczeństwa



Rysunek 1: "kropki" zamieszczane przez drukarki



Rysunek 2: model steganografii

- poufność
- autentyczność
- niezaprzeczalność
- integralność

Porównanie kryptografii i steganografii

	kryptografia	steganografia
cel	zapewnienie poufności	ukrycie komunikacji
obecność klucza	tak	opcjonalna
widoczność danych	nie	tak
modyfikacja struktury przetwarzanych danych	nie	tak

1.1 Słowniczek

- stegosystem - połączenie metod i narzędzi służących do tworzenia ukrytego kanału do przekazywania informacji
- wiadomość (payload) - przesyłane dane
- kontener/nośnik (carrier) - to wszelkie dane służące do ukrycia tajnej wiadomości
- stegokontener - dane i ukryta w nich tajna wiadomość
- kanał steganograficzny (stegochannel) - kanał transmisji stegokontenera
- klucz (stegokey) - tajny klucz potrzebny do ukrycia stegokontenera

1.2 Podział steganografii

Ze względu na kontener

- w plikach tekstowych
- w plikach audio

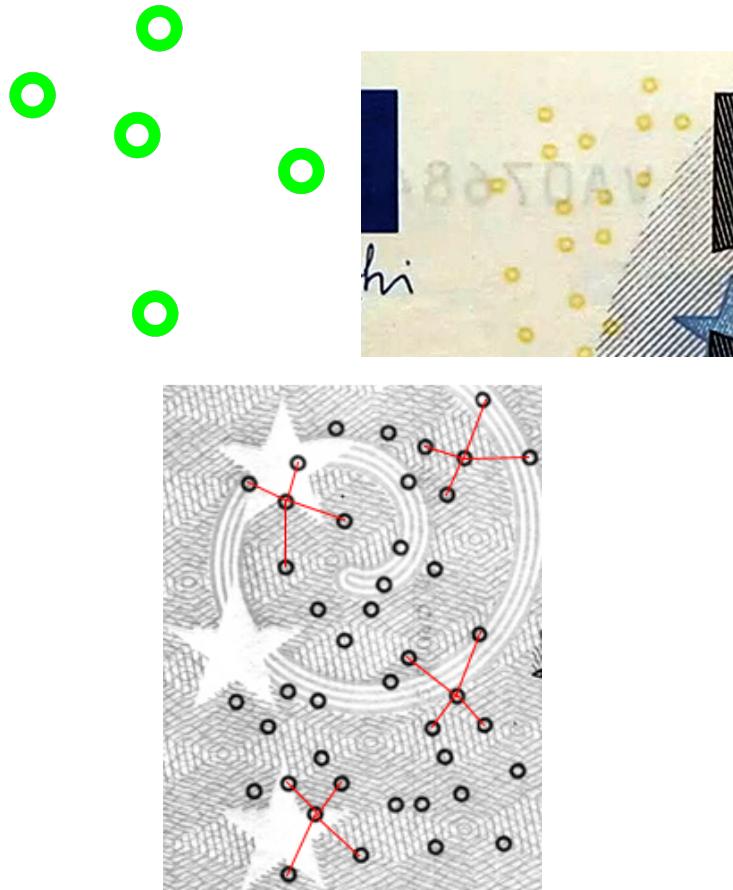
- w obrazach
- w ramkach różnych protokołów
- w plikach wykonawczych
- inne...

Ze względu na metodę modyfikacji nośnika

- **metody substytucji** - zamiana nadmiarowych danych nośnika
- **metody transformacyjne** - modyfikacja postaci falowej nośnika
- metody statystyczne - modyfikacja właściwości statystycznych nośnika
- metody generacji nośnika - ukrywanie informacji podczas tworzenia samego nośnika
- metody rozproszonego widma - ukrycie poprzez rozpraszanie danych
- metody zniekształcańcowe - wprowadzenie zniekształceń do nośnika i pozyskanie informacji poprzez porównanie nośnika oryginalnego i zniekształconego

2 Przykłady rzeczywistych zastosowań steganografii

2.1 Eurion



Rysunek 3: EURion, przykładowy układ na banknocie euro, dollarze

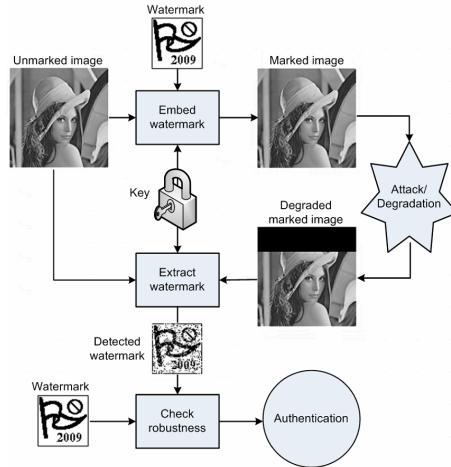
EURion jak i inne podobne zabiegi stanowią metodę przeciwdziałania fałszerstwom. Na banknotach umieszczane są zbiory kropek o różnych średnicach i względnych pozycjach (te parametry są sekretem). Kropki te tworzą fingerprint, który jest wykrywany przez oprogramowanie do skanowania (za pomocą metod detekcji wzorca) i wszelkie próby kopiowania banknotów są blokowane.

2.2 Znaki wodne

Umieszczanie znaków wodnych w plikach ma na celu zamieszczenie informacji o właściwym prawie autorskim. Wykorzystujemy różnych metod steganografii w celu zapewnienia:

- trudności w usunięciu
- odporności na transformacje (**robustness**)
- niedostrzegalności (**perceptibility**)
- przepustowość

z czego najważniejszą cechą jest odporność na transformacje



Rysunek 4: schemat zamieszczania znaku wodnego



Rysunek 5: CAP(Coded Anti Piracy) - przykład znaku wodnego zamieszczanego w filmach do identyfikacji źródła nielegalnych kopii

3 Przegląd technik steganografii

3.1 Modyfikacja LSB obrazu

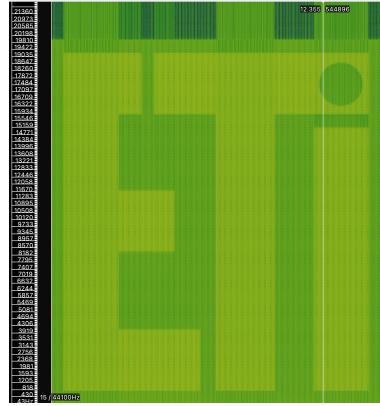
Jest to klasyczny algorytm steganografii, którego główną wadą jest łatwość w wykryciu/zniszczeniu wiadomości (np. przez wyzerowanie najmłodszych bitów). Przed niechcianym odczytem wiadomości możemy zapobiec poprzez zastosowanie kriptografii. Zasada działania algorytmu jest prosta:

1. wybierz, w którym kanale zapisać bity wiadomości (r,g,b, a, może obraz czarnobiały?)
2. zastąp stare wartości najmłodszych bitów określonego kanału obrazu kolejnymi bitami wiadomości

Analogiczna metoda jest możliwa na plikach dźwiękowych, tylko tam zmieniamy LSB próbek.

3.2 Gamma trick*

3.3 Ukrywanie obrazów w spektrogramach



Rysunek 6: Obraz zamieniony na dźwięk, bez żadnego "ukrywającego" pliku dźwiękowego, dźwięk ten jest odbierany jako szum

Inną z technik jest ukrywanie danych w określonych częstotliwościach pliku dźwiękowego. Jak to działa? Zaczynamy od konwersji obrazu na odcienie szarości. Następnie korzystamy ze wzoru (IDFT - inverse discrete fourier transform)

$$x = \sum_{y=0}^{H-1} I[x, y] \sin\left(\frac{2\pi f i}{S}\right)$$

gdzie:

H to wysokość obrazu,

x to próbka odpowiadająca x -tej kolumnie obrazu,

$I[x, y]$ to jasność piksela o współrzędnych x, y ,

S to częstotliwość próbkowania,

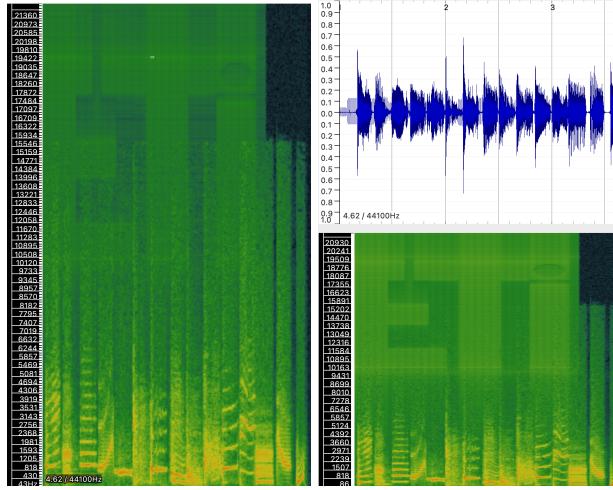
i to numer próbki w bloku (odwrotna transformacja Fouriera operuje na blokach stałego rozmiaru, które łączy w wartość wielomianu w punkcie x)
rozmiar bloku definiuje szerokość umieszczonego obrazu (w sensie ilości próbek)

$$f = y * \frac{f_{max} - f_{min}}{H} H + f_{min}$$

f_{min} to częstotliwość, od której zaczyna się dolna krawędź obrazu, f_{max} to górna krawędź (np. $f_{min} = 20$ Hz, $f_{max} = 20$ kHz). Chcąc dodać ukryty sygnał do istniejącego pliku dźwiękowego dodajemy sygnały

$$I' = I + \beta x$$

β to współczynnik tłumienia mający na celu ukrycie "brzęczenia" ukrytego obrazu



Rysunek 7: Z lewej: przytłumiony sygnał obrazu umieszczony w spektrogramie istniejącego pliku dźwiękowego, z prawej: ilustracja szumów przy braku tłumienia obrazu - szum tworzy widoczne gołykiem wąsy

3.4 Ukrywanie archiwów w obrazach

Kolejny bardzo prosty sposób na ukrycie danych w obrazach polega na sklejeniu dwóch plików ze sobą, uzyskując tym samym plik polyglot. Wymaga to jednak, żeby formaty akceptowały "śmieci" przed nagłówkiem. Przykładami takich formatów są pdf, rar, zip. Metodę tą można łatwo wykryć na przykład

za pomocą strings/binwalk.

Zapisywanie:

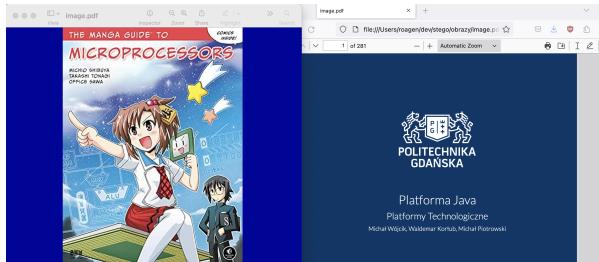
```
$ cat microprocessors.jpg JAVA_slajdy.pdf > image.jpg
```

W ten sposób ukrywać możemy zwykłe pliki tekstowe

```
$ cat microprocessors.jpg haslo.txt > image.jpg
```

wówczas

```
$ strings image.jpg
...
ri]P
:9w;
!`{?
haslo
```



Rysunek 8: Przykład pliku "poligloty", który może być jednocześnie interpretowany jako .jpg i .pdf

Szczególnym przypadkiem złączonych plików są pliki **GIFAR** - gif + jar. Przy nieodpowiednio zabezpieczonej stronie umożliwiającej umieszczanie gifów atakujący może uruchomić kod z pliku jar będącego częścią zamieszczonego gifa. Jary podobnie jak wszystkie formaty bazujące na formacie zip umożliwiają umieszczanie dodatkowych bajtów przed nagłówkiem.

3.5 Homoglyphy - Twitter Steganography

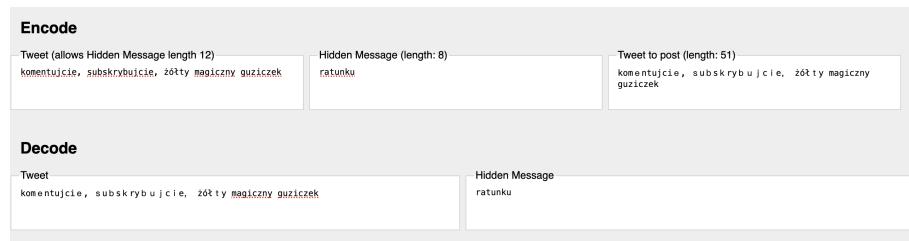
Homoglyphy to znaki, których kształty mogą być interpretowane na różne sposoby. Chcąc przy ich użyciu ukryć wiadomość musimy:

1. zdefiniować alfabet wiadomości

- ile bitów na znak? np. 6
- jak wygląda alfabet np. _abcdefghijklmnopqrstuvwxyz123456789

- dla powyższego alfabetu znak a będzie miał kod 000001 a np. 1001100
2. ustalić tekst wiadomości, która będzie kontenerem
 3. ustalić tekst ukrytej wiadomości i przekodować go na alfabet
 4. dla każdego znaku kontenera
 - (a) sprawdzamy ile ma homoglifów h
 - (b) liczbę różnych homoglifów danego znaku możemy zakodować na $\lceil \log_2 h \rceil = b$ bitach
 - (c) bierzemy b bitów ukrywanej wiadomości i na ich podstawie wybieramy, który z homoglifów zapiszemy do tekstu wyjściowego
 - (d) czyli na przykład, jeżeli pierwsza litera kontenera to A, które ma 4 homoglify, to bierzemy 2 bity wiadomości, jeżeli jest to 00 to znaku nie zmieniamy, 01 - wybieramy pierwszy homoglyph itd...

Żeby zdekodować wiadomość musimy znać alfabet i ilość bitów na znak.



Rysunek 9: przykład wiadomości zakodowanej za pomocą Twitter Steganography

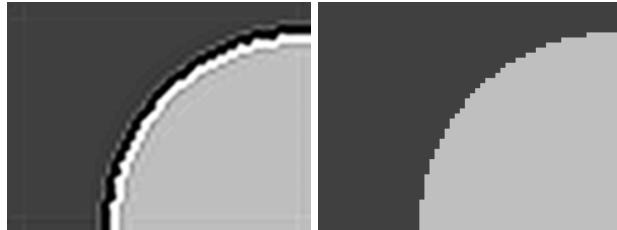
3.6 steghide

4 Steganoanaliza

Steganoanaliza (Steganalysis) jest tym samym dla steganografii, czym kryptoanaliza dla kryptografii - sztuką detekcji ukrytych wiadomości. Metody steganoanalityczne opierają się głównie na analizie parametrów statystycznych pliku. Przykładowe techniki

- analiza widma sygnału (przykładowo przeszukiwanie wysokich częstotliwości)
- niespójności w kompresji (np. "edge ringing" w kompresji JPEG)
- wykorzystywanie oryginalnego nośnika, jeżeli jest dostępny
- szukanie nietypowych wzorców

- użycie metod uczenia maszynowego



Rysunek 10: obraz z "edge ringing" i bez - jest to przewidywalne zniekształcenie; proste algorytmy steganografii mogą mieć problem z dobrym odwzorowaniem artefaktów o wysokim prawdopodobieństwie wystąpienia

4.1 Zadania steganoanalizy

- detekcja istnienia kanału steganograficznego (steganoanaliza pasywna)
- zniszczenie wiadomości w stegokontenerze
- ekstrakcja wiadomości ze stegokontenera

4.2 Problemy steganoanalizy

- wiele szyfrów ma taką właściwość, że produkuje szyfrogramy przypominające szum biały (o kompletnie płaskim widmie),
- barrage noise - bombardowanie stegokontenerami z losowymi/bezwartościowymi informacjami

4.3 Steganoanaliza LSB

Prosta metoda wykrycia, czy obraz zawiera zakodowaną wiadomość

1. dzielimy piksele na bloki
2. dla każdego bloku liczymy wartość średnia LSB

Jeżeli w obrazie ukryto **zaszyfrowaną** wiadomość, to niektóre regiony bloków powinny mieć średnią ≈ 0.5 (ze względu na losowy charakter szyfrów). Możliwe są oczywiście false positives, gdy piksele w niezmodyfikowanym obrazie mają taką charakterystykę.

5 Narzędzia steganoanalityczne

5.1 strings

strings to linuxowa komenda wypisująca łańcuchy znaków z danego pliku. Dzięki temu możemy pozyskać informacje o metadanych ukrytych np. w plikach wykonawczych

```
[roagen in ~/dev/rvtest λ strings a.out
riscv
rv32i2p0_m2p0
_boot
.symtab
.strtab
.shstrtab
.text
.data
.bss
.riscv.attributes
```

Rysunek 11: przykład użycia strings na pliku wykonawczym w architekturze rv32

5.2 binwalk

jest to program do znajdowania istniejących nagłówków plików w innym pliku, program umożliwia także analizę entropii pliku

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, JFIF standard 1.01
59235	0xE763	PDF document, version: "1.4"
107288	0x1A318	Zlib compressed data, default compression

Rysunek 12: przykład użycia binwalk na pliku poliglocie jpg + pdf

5.3 stegseek

5.4 StegExpose

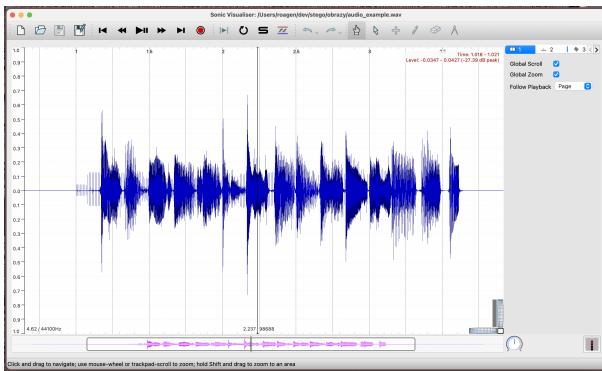
StegExpose to narzędzie do wykrywania steganografii LSB bazujące na metodzie RS, która jest podobna do wcześniej omówionej prostej steganoanalizy LSB, z wykorzystaniem teorii grup.

```
[roagen in ~/Program/StegExpose on master λ java -jar StegExpose.jar testFolder fast
stego_6664658261_e455d262b5_2.png is suspicious. Approximate amount of hidden data is 114785 bytes.
stego_6672108499_85c582a7ff.png is suspicious. Approximate amount of hidden data is 137947 bytes.
stego_6672542201_532f70bffe.png is suspicious. Approximate amount of hidden data is 67141 bytes.
```

Rysunek 13: przykład użycia StegExpose

5.5 Sonic Visualizer

Sonic Visualizer to program do wizualizacji plików dźwiękowych, który może być użyteczny do znajdowania podejrzanych szumów itp.



Rysunek 14: ilustracja wizualizacji sygnału z podejrzany szumem

6 Źródła

- https://pl.wikipedia.org/wiki/Steganografia_drukarkowa
- <https://royalprice.ru/pl/setting/steganografiya-i-stegoanaliz-obzor-sushchestvuyushchih-programm-i-algoritmov/>
- https://www.researchgate.net/figure/The-model-of-steganography-and-steganalysis_fig1_333772050
- <http://datagenetics.com/blog/september12015/index.html>
- <https://holloway.nz/steg/>
- [https://en.wikipedia.org/wiki/Polyglot_\(computing\)](https://en.wikipedia.org/wiki/Polyglot_(computing))
- <https://github.com/livz/cloacked-pixel>
- <https://github.com/b3dk7/StegExpose>
- https://link.springer.com/chapter/10.1007/11424826_54
- <https://github.com/fallais/tweg>
- <http://datagenetics.com/blog/september12015/index.html>
- <https://carlmastrangelo.com/blog/gamma-steganography>