

Demystyfikacja steganografii

Marek Dalewski

14 kwietnia 2017

Spis treści

1 Wstęp	1
2 Czym jest steganografia?	1
3 Notka o budowie obrazów	2
4 Algorytm ukrywający dane	2
5 Demonstracja algorytmu	4

Wstęp

Niniejszy tekst powstał jako wprowadzenie dla osób nie nieposiadających technicznego wykształcenia, chcących zapoznać się z tematyką ukrywania cyfrowych informacji - steganografii cyfrowej.

Czym jest steganografia?

To nauka o ukrywaniu faktu prowadzenia komunikacji.

To tyle - powyższy akapit w pełni oddaje cały zakres steganografii, jest sformułowany prostym i przystępny językiem, a jednak nie pozostawia czytelnika usatysfakcjonowanego. Dzieje się tak dlatego, że opis jest zbyt abstrakcyjny, a dokładniej operuje nieznaną czytelnikowi abstrakcją. Bo jak komunikacja jest ukrywana? Jaka komunikacja? I co to w ogóle znaczy ukrywać komunikację?

Jak więc wytłumaczyć to pojęcie? Może studium uproszczonego przypadku? Tak, by dać praktyczny przykład bez stawiania wymagań specyficznej wiedzy technicznej. Wydaje się to dobry pomysł, jednak na ogół wprowadzanie owych uproszczeń usuwa z przykładu to, co dla odbiorcy najistotniejsze. Odbiorca, posiadający wymaganą wiedzę techniczną, czy też nie, chce zapoznać się z tematem z jakiegoś powodu i na ogół przyczyna jest znacznie mniej błaha niż zaspokojenie ciekawości. Problem jednak w tym, jak wytłumaczyć temat, czy chociaż do niego wprowadzić czytelnika w obliczu postawionego wcześniej założenia nie posiadania wiedzy technicznej. Cóż, pozostaje tylko spróbować wytłumaczyć możliwie przystępnym językiem pełny, a nie uproszczony przykład.

Zacznijmy jednak od przedstawionej wcześniej definicji. Według niej mamy tu do czynienia z komunikacją i chodzi o jakąkolwiek komunikację - wymianę tekstów albo na

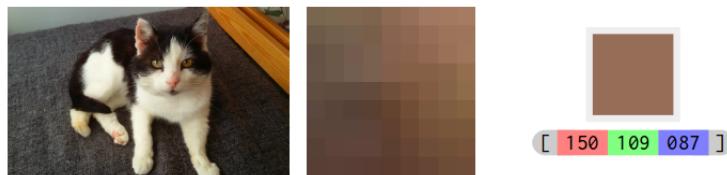
przykład obrazów między dwiema osobami. W dalszym przykładzie wykorzystamy właśnie obrazy. Osoby mogą używać dowolnego medium komunikacyjnego, na przykład poczty e-mail. Nadal jednak nie wytłumaczyliśmy pojęcia ukrywania. Co to ma oznaczać? Uciekniemy się do prostej sztuczki - by ukryć jedną komunikację użyjemy innej. Oryginalny komunikat ukryjemy w szumie komunikacyjnym, który sami wygenerujemy. Jak będzie to dokładnie wyglądać w przykładzie? Ukryjemy nasz oryginalny komunikat (oryginalną wiadomość) - obraz w innym obrazie.

Zanim jednak do tego przejdziemy, musimy wytłumaczyć kilka niezbędnych pojęć na temat budowy obrazów cyfrowych.

Notka o budowie obrazów

By przejść do omawiania samego algorytmu ukrywania obrazu w obrazie, musimy ustalić wspólny poziom wiedzy co do budowy samych obrazów. To, co trzeba wiedzieć, to że obrazy składają się z pikseli. Z kolei piksel to taki jednokolorowy kwadracik¹, a kolor jednoznacznie charakteryzuje trzy liczby². Każda z zakresu od 000 do 255 (każda składa się z trzech cyfr). Odpowiadają one natężeniu odpowiednio barwy czerwonej, zielonej i niebieskiej (rysunek 1).

W skrócie: piksel, podstawowy atom obrazu, jest kwadracikiem jednego koloru i można go scharakteryzować przy użyciu trzech liczb z zakresu od 000 do 255.



Rysunek 1: Obraz i jego piksele

Warto też zwrócić uwagę na różnicę między kolorami w miarę zwiększania między nimi odległości³ - jak niewielkie zachodzą różnice wizualne (rysunek 2).

Algorytm ukrywający dane

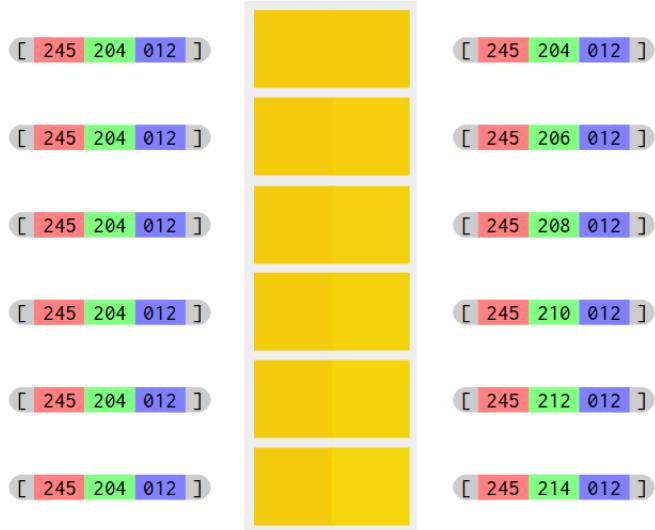
Po zapoznaniu się z budową obrazów cyfrowych, nie pozostaje już nic innego, niż przejść do metody steganograficznej i rozpocząć ukrywanie danych. Sposób postępowania jest następujący⁴:

¹Na potrzeby niniejszej dyskusji całkowicie pominiemy dokładną definicję piksela i fakt, że piksel niekoniecznie musi być kwadratowy. Ponadto warto zaznaczyć, że - zależnie od formatu graficznego - piksele są przetwarzane przed zapisaniem do pliku (na przykład przez algorytm stratnej albo bezstratnej kompresji).

²By nie zakłócać przykładu niepotrzebnymi informacjami, pominiemy tu fakt, że istnieją inne reprezentacje przestrzeni barw, niż użyta tu 24 bitowa reprezentacja RGB.

³Termin "zwiększenie odległości między kolorami" może budzić wiele wątpliwości. Zauważmy, że możemy interpretować kolor z palety barw RGB jako punkt w trójwymiarowej przestrzeni. Gdy mamy dwa kolory, tj. dwa punkty w przestrzeni, możemy mówić o odległości między nimi.

⁴Stosowane nazewnictwo obrazów jest następujące: "obraz z wiadomością" to obraz, który **jest ukrywany**; "obraz oryginalny" to obraz, **w którym ukrywamy**.

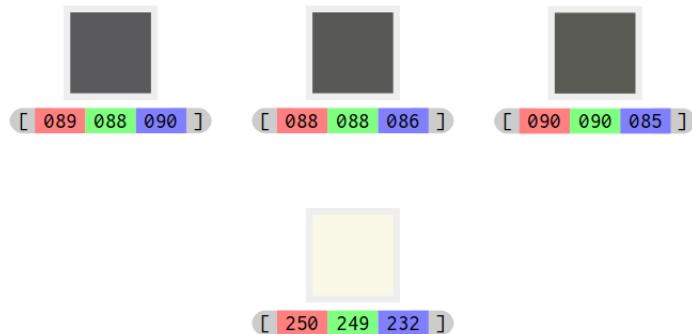


Rysunek 2: Różnice wizualne między kolorami, w miarę zwiększania odległości między nimi

1. Wybieramy piksel obrazu z ukrywaną wiadomością i odpowiadające mu trzy piksele w oryginalnym obrazie⁵ (rysunek 3).
2. Zerujemy cyfrę jedności dla każdego kanału koloru, dla każdego piksela z obrazu oryginalnego (rysunek 4).
3. Zgodnie z rysunkiem 5 wstawiamy cyfry z piksela obrazu z wiadomością do pikseli obrazu oryginalnego.

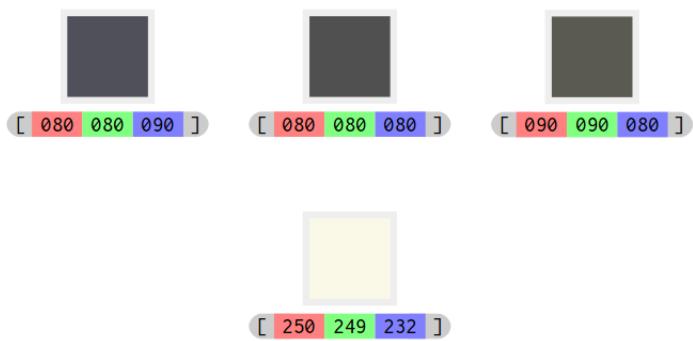
Powyższe czynności powtarzamy dla wszystkich pikseli w obrazie z wiadomością.

By wyeksportować ukryte wcześniej dane i odzyskać oryginalny obraz z wiadomością, wystarczy odwrócić przedstawione kroki.

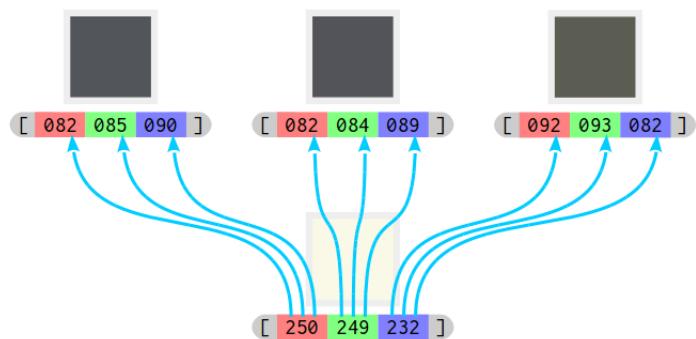


Rysunek 3: Piksel obrazu z wiadomością wraz z odpowiadającymi mu pikselami oryginalnego obrazu

⁵Obraz oryginalny musi być co najmniej trzy razy szerszy, by każdy piksel obrazu z wiadomością znalazł odpowiadające mu trzy piksele obrazu oryginalnego.



Rysunek 4: Zerowanie jedności w pikselach obrazu oryginalnego

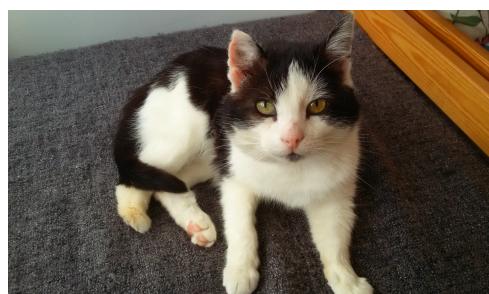


Rysunek 5: Schemat wstawiania cyfr

Powyższa metoda jest bardzo prosta, a zarazem jest to jeden z najczęściej spotykanych algorytmów steganograficznych. Jego wadą jest łatwość wykrycia, gdy tylko znamy metodę ukrywania. Istnieją oczywiście dużo bardziej zaawansowane i bezpieczniejsze metody. Bazują one jednak na zaawansowanych przekształceniach matematycznych, których przytaczanie, na potrzeby krótkiego wprowadzenia do tematu jest bezzasadne.

Demonstracja algorytmu

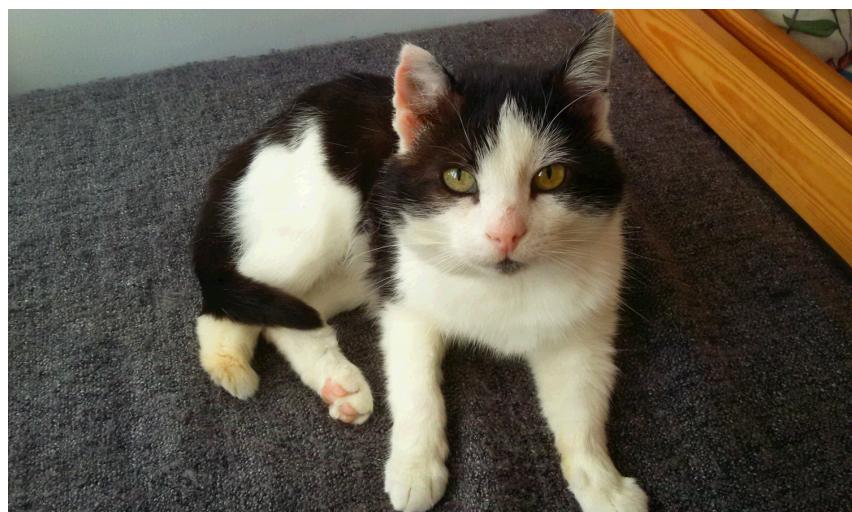
Na rysunku 8 możemy zobaczyć rezultat działania powyższego algorytmu dla obrazu oryginalnego (rysunek 6) i obraz z wiadomością (rysunek 7). Zwróćmy uwagę, że zmian wprowadzonych działaniem algorytmu nie widać gołym okiem.



Rysunek 6: Obraz oryginalny



Rysunek 7: Obraz z wiadomością do ukrycia



Rysunek 8: Wynik działania algorytmu