Biuletyn WAT
Vol. LXV, Nr 4, 2016

Wykorzystanie losowych kodów liniowych w steganografii

KAMIL KACZYŃSKI

Wojskowa Akademia Techniczna, Wydział Cybernetyki, Instytut Matematyki i Kryptologii,
00-908 Warszawa, ul. gen. S. Kaliskiego 2, kamil.kaczynski@wat.edu.pl

Streszczenie. Kodowanie syndromami kodów liniowych jest zabiegiem pozwalającym na poprawę parametrów algorytmów steganograficznych. Zastosowanie losowych kodów liniowych pozwala na dużą elastyczność w doborze parametrów kodu liniowego, jednocześnie pozwalając w prosty sposób tworzyć macierze kontroli parzystości. W niniejszym artykule przedstawiona została modyfikacja algorytmu zastępującego LSB pikseli wykorzystująca losowy kod liniowy [8, 2]. Zaproponowano wzorcową implementację opracowanego algorytmu oraz praktyczną ocenę jego parametrów wykonaną na bazie obrazów testowych.
Słowa kluczowe: steganografia, losowe kody liniowe, LSB, RLC
DOI: 10.5604/12345865.1228962

1. Wstęp

Algorytmy steganograficzne wykorzystujące obrazy rastrowe należą do najbardziej popularnych. Wykorzystanie części redundantnych danych występujących w obrazach cyfrowych pozwala na ukrycie danych cyfrowych w sposób, który uniemożliwia proste ich wykrycie. Umożliwia to utworzenie kanału komunikacji dostępnego jedynie dla upoważnionego odbiorcy, z pominięciem możliwości przejęcia przesyłanych danych przez napastnika.

Algorytm modyfikujący najmniej znaczące bity (LSB) piksela obrazu to popularna i dobrze zbadana metoda steganograficzna. Podstawą działania jest zmiana najmniej znaczącego bitu wybranej barwy piksela na wartość zgodną z bitem ukrywanej informacji. Algorytm ten zachowuje wysoką jakość obrazu źródłowego,

co wyklucza zastosowanie w procesie jego wykrywania metod analizy organoleptycznej.

Oczekuje się, aby algorytm steganograficzny wykazywał się jak najwyższą pojemnością przy jednoczesnym zachowaniu odpowiedniego stopnia niewykrywalności ukrytego przekazu. Pojemność algorytmu jest określana poprzez tzw. współczynnik osadzania, który określa liczbę bitów możliwych do ukrycia w jednym bicie nośnika. Poprzez niewykrywalność ukrytego przekazu należy rozumieć zdolność przekazywania wiadomości w sposób, który nie może być wykryty organoleptycznie ani poprzez analizę własności statystycznych. Każde wprowadzenie dodatkowych treści do obrazu zmienia jego własności statystyczne, stąd istotne jest zmniejszanie liczby zmian wprowadzanych do nośnika. Przyjmuje się zatem, że wraz ze zwiększaniem liczby wprowadzanych zmian znacząco wzrasta możliwość wykrycia faktu wprowadzenia ukrytej informacji. Liczba wprowadzanych zniekształceń dla danego algorytmu jest określana jako tzw. średnie zniekształcenie określające liczbę zmian wprowadzanych do nośnika w celu ukrycia jednego symbolu. Na podstawie współczynnika osadzania i średniego zniekształcenia możliwe jest określenie tzw. sprawności osadzania, będącej ilorazem średniego zniekształcenia i współczynnika osadzania. Parametr ten definiuje możliwą liczbę ukrytych bitów przy wprowadzeniu do nośnika jednej zmiany.

2. Algorytm steganograficzny

Poprzez losowy kod liniowy $[n, k]$ rozumiemy rodzaj blokowego kodu korekcji błędów, który posiada macierz kontroli parzystości $H = [I_{n-k}, D]$, gdzie I_{n-k} jest macierzą jednostkową o wymiarach $(n-k) \times (n-k)$, zaś D jest macierzą o wymiarach $(n-k) \times k$, o elementach wybranych pseudolosowo z GF(2). Zastosowanie losowych kodów liniowych pozwala na wytwarzanie algorytmów steganograficznych o płynnie dobieranym współczynniku osadzania. Dodatkowo, taki kod może cechować się krótką długością bloku, co w znaczący sposób zmniejsza złożoność obliczeniową zmodyfikowanych z jego wykorzystaniem algorytmów steganograficznych.

W odróżnieniu od innych liniowych kodów korekcyjnych, macierz kontroli parzystości kodów losowych jest wygenerowana w sposób pseudolosowy. W związku z tym podstawowa dla procesu kodowania syndromami czynność odnajdywania liderów warstw staje się problemem NP trudnym. Najprostszym i najbardziej efektywnym sposobem jego rozwiązania jest wygenerowanie tablicy liderów warstw dla każdego z możliwych syndromów i jej wykorzystywanie w procesie ukrywania i wyodrębniania danych.

Na potrzeby modyfikowanego algorytmu wytworzona została macierz kontroli parzystości dla kodu [8, 2] o następującej postaci:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{1}$$

Kod posiadający taką macierz kontroli parzystości ma następujący zbiór liderów warstw odpowiadających syndromom:

Tabela 1

Liderzy warstw kodu losowego [8, 2]

Syndrom	Lider warstwy	Syndrom	Lider warstwy	Syndrom	Lider warstwy	Syndrom	Lider warstwy
000000	00000000	010000	01000000	100000	10000000	110000	11000000
000001	00000100	010001	01000100	100001	10000100	110001	11000100
000010	00001000	010010	01001000	100010	10001000	110010	11001000
000011	00001100	010011	01001100	100011	10001100	110011	11001100
000100	00010000	010100	01010000	100100	10010000	110100	11010000
000101	00010100	010101	01010100	100101	10010100	110101	11010100
000110	00011000	010110	01011000	100110	10011000	110110	11011000
000111	00011100	010111	01011100	100111	10011100	110111	11011100
001000	00100000	011000	01100000	101000	10100000	111000	11100000
001001	00100100	011001	01100100	101001	10100100	111001	11100100
001010	00101000	011010	01101000	101010	10101000	111010	11101000
001011	00101100	011011	01101100	101011	10101100	111011	11101100
001100	00110000	011100	01110000	101100	00000010	111100	01000010
001101	00000001	011101	01000001	101101	00000110	111101	01000110
001110	00111000	011110	01111000	101110	00001010	111110	01001010
001111	00001001	011111	01001001	101111	00001110	111111	01001110

Algorytm ukrywania wiadomości w LSB pikseli danego koloru z wykorzystaniem wyżej wymienionego kodu liniowego [n, k] przebiega według poniższych kroków:

1. Odczytujemy kolejne n bitów nośnika, tworząc wektor o wymiarach [1, n].
2. Obliczamy iloczyn macierzowy Hx.
3. Pobieramy kolejne n-k bitów ukrywanej wiadomości, tworząc wektor m o wymiarach [1, $n - k$].

4. Obliczamy $s = m - Hx$.
5. Dla obliczonego s wyszukujemy w tabeli 1 odpowiadającego lidera warstwy — wektora e.
6. Tworzymy wektor $y = x + e$.
7. Zamieniamy wektor x nośnika na wektor y. Jeżeli pozostały jeszcze nieukryte bity wiadomości, przechodzimy do punktu 1.

Dla przedstawionego algorytmu z kodem [8, 2] istnieje możliwość ukrycia $2^6 = 64$ różnych wiadomości binarnych. Wektor kodowy składa się z 8 symboli. Ukrycie wiadomości wymaga zmodyfikowania wektora kodowego w taki sposób, aby jego syndrom był równy ukrywanej wiadomości. Na podstawie tabeli 1 można zatem określić oczekiwaną liczbę zmian wprowadzonych do nośnika:

$$R = 0 \cdot \frac{1}{64} + 1 \cdot \frac{8}{64} + 2 \cdot \frac{20}{64} + 3 \cdot \frac{22}{64} + 4 \cdot \frac{11}{64} + 5 \cdot \frac{2}{64} = \frac{21}{8}. \tag{2}$$

Powyższe pozwala na określenie średniego zniekształcenia:

$$D = \frac{\frac{21}{8}}{8} = \frac{21}{64}. \tag{3}$$

Współczynnik osadzania jest równy:

$$E = \frac{n-k}{n} = \frac{8-2}{8} = \frac{3}{4}. \tag{4}$$

Opierając się na wyznaczonych wartościach współczynnika osadzania oraz średniego zniekształcenia, sprawność osadzania jest dana poniżej:

$$\frac{E}{D} = \frac{\frac{3}{4}}{\frac{21}{64}} = \frac{16}{7} \approx 2{,}286. \tag{5}$$

W poniższej tabeli dokonano porównania parametrów proponowanego algorytmu z algorytmami z [1] (LSB_Hamming), [3] (PM1_Ternary), a także z podstawowym algorytmem LSB.

Tabela 2

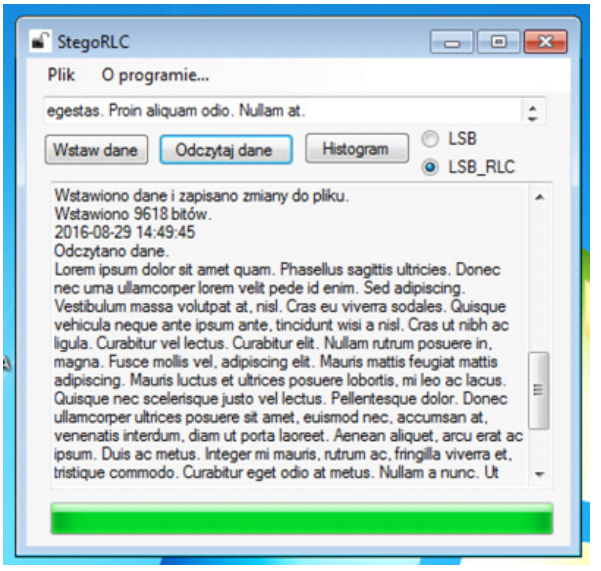
Porównanie parametrów algorytmów steganograficznych

	LSB_RLC	LSB_Hamming	PM1_Ternary	LSB
Średnie zniekształcenie	$\frac{21}{64}$	$\frac{1}{8}$	$\frac{2}{27}$	$\frac{1}{2}$
Współczynnik osadzania	$\frac{3}{4}$	$\frac{3}{7}$	$\frac{2\log_2 27}{26} \approx 0{,}366$	1
Sprawność osadzania	$\frac{16}{7} \approx 2{,}286$	$\frac{24}{7} \approx 3{,}429$	$\frac{27\log_2 27}{26} \approx 4{,}938$	2

3. Implementacja

W celu praktycznego sprawdzenia wykonanych założeń oraz określenia stopnia bezpieczeństwa algorytmu, dokonana została jego wzorcowa implementacja. Wykorzystano w tym celu środowisko Microsoft Visual Studio 2015 oraz język programowania C#.

Przyjęto założenia, że formatem plików, na którym będzie operowała aplikacja, będzie BMP z 24-bitową głębią koloru. Kanał przenoszący ukrytą treść to kanał



Rys. 1. Aplikacja StegoRLC

Źródło: opracowanie własne

koloru niebieskiego. Aplikacja pozwala na ukrycie dowolnego ciągu binarnego, jednak ze względów pozwalających na uproszczenie procedury testowania założono, że ukrywaną informacją będzie tekst w kodowaniu UTF-8.

Aplikacja umożliwia zastosowanie szyfrowania ukrywanej wiadomości algorytmem AES, pozwala także na losowe rozmieszczenie danych wewnątrz obrazu.

4. Stegoanaliza

Zaproponowany w niniejszej pracy algorytm jest modyfikacją algorytmu modyfikującego najmniej znaczące bity pikseli (LSB). W związku z powyższym, do przeprowadzenia analizy jego bezpieczeństwa zostaną wykorzystane podstawowe metody mające zastosowanie w wykrywaniu komunikacji prowadzonej poprzez algorytm niezmodyfikowany.

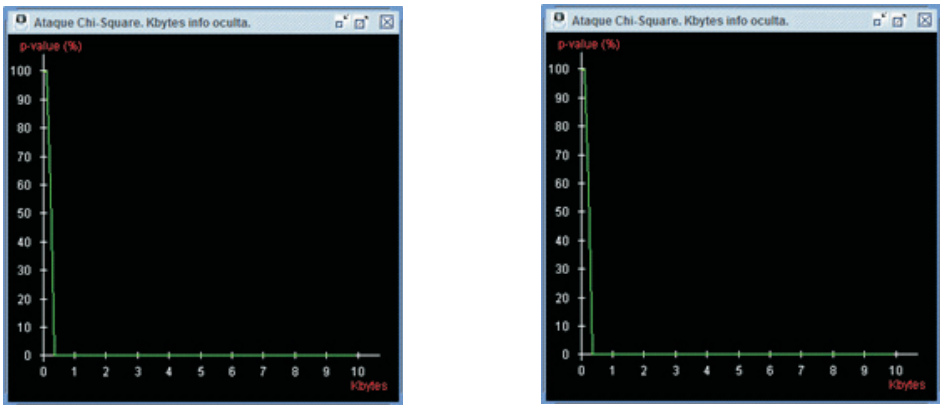
W celu przeprowadzania badań przyjęto, że zostaną wykorzystane obrazy z publicznie dostępnej bazy danych, każdy o wymiarach 300 × 200 pikseli. W obrazach zostanie ukryty tekst, którego rozmiar będzie równy 5625 bajtów, co będzie oznaczało całkowite wykorzystanie pojemności nośnika. W dalszej części pracy obrazy oryginalne będą nazywane *obraz*, natomiast obrazy zmodyfikowane *obraz_stego*.

Rys. 2. Obrazy arctichare, sails i tulips
Źródło: domena publiczna

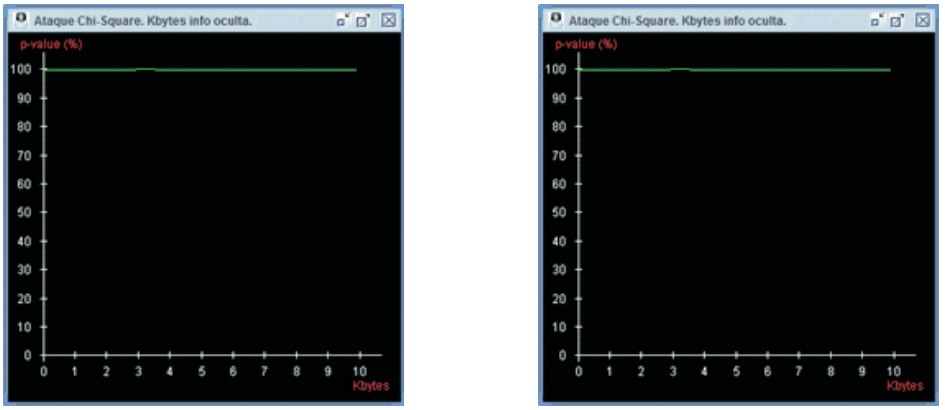
Dla algorytmów zmieniających najmniej znaczący bit ataki wizualne nie mają zastosowania w przypadku braku oryginalnego obrazu. Do stegoanalizy zastosowano zatem algorytmy statystyczne — atak Chi-kwadrat [5] oraz stegoanalizę RS [7, 8]. W przypadku metody Chi-kwadrat nie udało się odróżnić obrazu oryginalnego od zmodyfikowanego. Na rysunkach 3, 4 i 5 przedstawiono porównanie dla każdego z trzech obrazów testowych.

Analiza powyższych wyników wskazuje na brak istotnych różnic pomiędzy obrazem oryginalnym a zmodyfikowanym. Pozwala to wyciągnąć wniosek, że zaproponowany algorytm jest transparentny dla stegoanalizy metodą Chi-kwadrat.

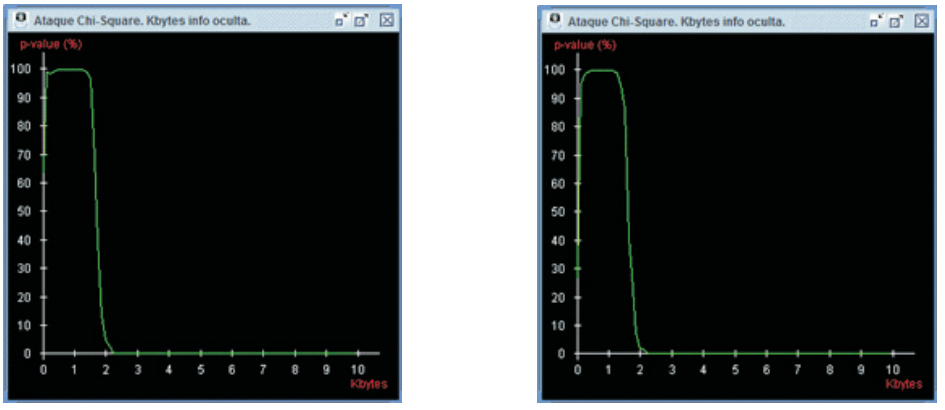
Kolejnym etapem badania poziomu bezpieczeństwa proponowanego algorytmu było wykorzystanie stegoanalizy RS [7, 8]. Ten typ ataku pozwala bardzo dokładnie



Rys. 3. Wynik analizy Chi-kwadrat dla obrazu arctichare (po lewej) i artcitchare_stego (po prawej)



Rys. 4. Wynik analizy Chi-kwadrat dla obrazu sails (po lewej) i sails_stego (po prawej)



Rys. 5. Wynik analizy Chi-kwadrat dla obrazu tulips (po lewej) i tulips_stego (po prawej)

określić fakt wykorzystania algorytmów modyfikujących LSB obrazu. W poniższej tabeli zamieszczono porównanie wskazywanych przez atak RS wartości względnych zmienionych w obrazie pikseli.

Tabela 3

Wyniki stegoanalizy RS

	Obraz oryginalny	Stegogram
artichare	6,01%	68,18%
sails	13,67%	73,79%
tulips	4,07%	73,88%

Dla stegoanalizy RS progiem, przy którym można uznać, że obraz przenosi ukrytą treść, jest ok. 8% zmienionych pikseli. Z powyższej tabeli wynika, że zaproponowany algorytm nie jest odporny na ten rodzaj ataku. Wynika to z wysokiej wartości średniego zniekształcenia dla proponowanego algorytmu. W celu praktycznego jego wykorzystania algorytm został zmodyfikowany z użyciem metody kompresji z [4]. Poniższa tabela zawiera zestawienie wyników.

Tabela 4

Zestawienie wyników analizy algorytmem RS

	Obraz oryginalny	Stegogram (algorytm oryginalny)	Stegogram (algorytm zmodyfikowany)
artichare	6,01%	68,18%	25,88%
sails	13,67%	73,79%	2,11%
tulips	4,07%	73,88%	6,09%

Zastosowanie modyfikacji z [4] pozwoliło na wyeliminowanie negatywnego wpływu średniego zniekształcenia na bezpieczeństwo proponowanego algorytmu. Jedynie w przypadku obrazu *arctichare* zwracany wynik przekracza próg alarmowy — jest to wynik specyfiki obrazu posiadającego bardzo duże obszary o jednolitej białej barwie. Tego typu obrazy nie są dobrym nośnikiem dla ukrytych treści i w praktyce nie są do tego celu wykorzystywane.

5. Wnioski

W niniejszej pracy zaproponowano modyfikację algorytmu LSB z wykorzystaniem losowych kodów liniowych. Zastosowanie kodowania syndromami tych kodów pozwala na poprawę wszystkich parametrów algorytmu steganograficznego,

co przekłada się zarówno na zwiększenie pojemności, jak i na obniżenie liczby wprowadzanych zniekształceń.

Algorytmy steganograficzne stosujące losowe kody liniowe cechują się dużymi możliwościami dostosowania parametrów do wymagań użytkownika, w szczególności szerokie są możliwości doboru współczynnika osadzania. W niniejszej pracy zaproponowano kod [8, 2], który gwarantował wysoką pojemność nośnika, wartości te mogą być jednak dobierane w sposób odpowiedni dla poszczególnych zastosowań. Podstawową wadą wykorzystania losowych kodów liniowych jest wysoka złożoność procesu określania liderów warstw — za najszybszą nadal uznaje się metodę pełnego przeszukania. W związku z tym parametry kodu należy dobierać w sposób, który zagwarantuje zadowalający poziom wydajności.

Zaproponowany algorytm cechuje się wysokim poziomem bezpieczeństwa, w podstawowej wersji jest odporny na ataki wizualne oraz stegoanalizę Chi-kwadrat. Uzyskanie odporności na stegoanalizę RS wymagało wykorzystania funkcji kompresji z [4]. Tak zmodyfikowany algorytm może zostać z powodzeniem wykorzystany do wytworzenia tajnego kanału łączności, np. poprzez popularne komunikatory internetowe.

Praca finansowana z działalności statutowej uczelni — praca RMN nr 820/2016 „Zastosowanie macierzy kontroli parzystości w tworzeniu algorytmów steganograficznych".

Artykuł wpłynął do redakcji 28.10.2016 r. Zweryfikowaną wersję po recenzjach otrzymano 9.11.2016 r.

LITERATURA

[1] Kaczyński K., *Steganografia z wykorzystaniem optymalnych kodów liniowych*, [w:] *Nowe techniki badań kryminalistycznych a bezpieczeństwo informacji*, red. nauk. B. Hołyst, J. Pomykała, P. Potejko, Wyd. PWN, 2014, 110-120.

[2] Gawinecki J., Kaczyński K., *Stegodroid — aplikacja mobilna do prowadzenia ukrytej komunikacji*, Studia Bezpieczeństwa Narodowego, r. 4, nr 6, Warszawa, 2014, 139-150.

[3] Kaczyński K., *Algorytm steganograficzny PM1 wykorzystujący trójkowy kod Hamminga*, Biuletyn WAT, vol. 64, nr 4, Warszawa, 2015, 257-267.

[4] Kaczyński K., *Zmniejszenie wrażliwości zmodyfikowanego algorytmu LSB na wybrane ataki statystyczne*, Biuletyn WAT, vol. 63, nr 4, Warszawa, 2014, 223-232.

[5] Westfeld A., Pfitzmann A., *Attacks on Steganographic Systems*, Lecture Notes in Computer Science, vol. 1768, Berline Heidelberg New York, 2000, 61-75.

[6] Marçal A.R.S., Pereira P.R., *A steganographic method for digital images robust to RS steganalysis*, International Conference Image Analysis and Recognition, Springer Berlin Heidelberg, 2005.

[7] Fridrich J., Goljan M., Rui Du, *Detecting LSB steganography in color, and grayscale images*, IEEE Multimedia, 8, 2001, 22-28.

[8] Fridrich J., Goljan M., Hogea D., Soukal D., *Quantitative steganalysis of digital images: estimating the secret message length*, Multimedia Systems, 9, 2003, 288-302.

[9] Qian Mao, *A fast algorithm for matrix embedding steganography*, Digital Signal Processing, 25, 2014, 248-254.

K. KACZYŃSKI

Random linear codes in steganography

Abstract. Syndrome coding using linear codes is a technique that allows improvement in the steganographic algorithms parameters. The use of random linear codes gives a great flexibility in choosing the parameters of the linear code. In parallel, it offers easy generation of parity check matrix. In this paper, the modification of LSB algorithm is presented. A random linear code [8, 2] was used as a base for algorithm modification. The implementation of the proposed algorithm, along with practical evaluation of algorithms' parameters based on the test images was made.
Keywords: steganography, random linear codes, RLC, LSB
DOI: 10.5604/12345865.1228962