Biuletyn WAT
Vol. LXIII, Nr 4, 2014

Zmniejszenie wrażliwości zmodyfikowanego algorytmu LSB na wybrane ataki statystyczne

KAMIL KACZYŃSKI

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

Streszczenie. Algorytm LSB to jeden z najlepiej zbadanych algorytmów steganograficznych. Istnieje kilka typów ataków, które pozwalają na wykrycie faktu prowadzenia komunikacji z wykorzystaniem tego algorytmu, jest to m.in. atak RS oraz chi-kwadrat. W niniejszej pracy przedstawiono modyfikację algorytmu LSB, która wprowadza mniej zmian do nośnika niż algorytm oryginalny, a także poprzez wykorzystanie funkcji kompresji w znaczący sposób utrudnia wykrycie faktu wprowadzenia informacji do obrazu. W pracy zawarty został także opis głównych metod stegoanalitycznych wraz z ich zastosowaniem do zmodyfikowanego algorytmu LSB.
Słowa kluczowe: steganografia, kody cykliczne, LSB, BCH, chi-kwadrat, stegoanaliza
DOI: 10.5604/12345865.1131499

1. Wstęp

Łatwa dostępność oprogramowania pozwalającego na ukrywanie wiadomości w bitmapach spowodowała zwiększenie zainteresowania wykorzystywaniem steganografii do prowadzenia ukrytej komunikacji. Z dobrodziejstw steganografii korzystają nie tylko osoby, które przebywają na terenie krajów, w których wykorzystanie kryptografii do prowadzenia komunikacji jest zakazane, ale także zorganizowane grupy przestępcze oraz terroryści.

Powyższe wymogło wytworzenie rozwiązań pozwalających na odnalezienie podejrzanych treści w natłoku przesyłanych przez sieć Internet multimediów. Odpowiedzią dla algorytmu LSB były atak chi-kwadrat opracowany przez Westfelda i Pfitzmanna [1] oraz stegoanaliza RS autorstwa J. Fridricha [2, 3].

Naturalnym następstwem odkrycia powyższych ataków jest modyfikacja algorytmów steganograficznych w taki sposób, aby ataki te nie miały dalszego zastosowania. W poniższej pracy zaprezentowano podejście będące rozwinięciem zaprezentowanego w [4] na obrazy kolorowe, dla algorytmu LSB wykorzystującego tzw. kodowanie macierzowe z wykorzystaniem cyklicznych kodów korekcji błędów [5].

2. Algorytm steganograficzny

Działanie prostego algorytmu LSB polega na zamianie najmniej znaczących bitów nośnika na bity wiadomości. Ze względu na niedoskonałości ludzkiego zmysłu wzroku, wprowadzenie takiej zmiany jest niezauważalne, zarówno dla obrazów o pełnej palecie barw RGB (24 bity), jak i obrazów w odcieniach szarości (8 bitów). Taka konstrukcja algorytmu powoduje, że liczba wprowadzanych do nośnika zmian jest w przybliżeniu równa połowie liczby wszystkich ukrywanych bitów.

Rozwiązaniem problemu wprowadzania tak dużej liczby zmian jest tzw. kodowanie syndromami. W zaproponowanym algorytmie wykorzystano syndromy kodu BCH(15,7). Pozwala to na ukrycie 8 bitów danych w bloku 15-bitowym przy zmianie zera, jednego, dwóch lub trzech bitów nośnika. W przypadku pesymistycznym ukrycie 8 bitów informacji będzie wymagało zmiany 3 bitów nośnika, co w odniesieniu do zmiany 8 bitów dla prostego algorytmu LSB pozwala na znaczne zmniejszenie wprowadzanych zniekształceń. Zasada działania algorytmu jest przedstawiona poniżej.

Załóżmy, że blok danych nośnika to $V = \{v_0, v_1,\ldots, v_{14}\}$, natomiast blok danych po dokonaniu modyfikacji to $R = \{r_0, r_1,\ldots, r_{14}\}$. Oznaczmy ukrywaną wiadomość jako $m = \{m_0, m_1,\ldots, m_7\}$. Wiadomość może zostać obliczona z poniższej zależności:

$$m = R \times H^T,$$

gdzie macierz H^T to:

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

Proces wbudowania wiadomości m polega na odnalezieniu takiego R, że $R \times H^T = m$. Różnica pomiędzy blokiem V i R będzie oznaczana jako $E = \{e_0, e_1, \ldots, e_{14}\}$. Wektor ten wskazuje położenie bitów, które należy zmodyfikować, aby możliwe było ukrycie wiadomości.

$$E = V - R.$$

Ostatnim krokiem ukrywania wiadomości m w nośniku jest dodanie wektora błędu E do bloku R danych nośnika. Przedstawiony powyżej algorytm może zostać wykorzystany także do modyfikacji innych algorytmów steganograficznych.

Taka modyfikacja algorytmu LSB wprowadza do nośnika zmiany, które mogą zostać z łatwością wykryte z wykorzystaniem stegoanalizy statystycznej. Jeżeli oznaczymy obraz z ukrytą treścią jako y_1, a algorytm wprowadzania wiadomości jako funkcję E, x jako obraz nośnik, a m jako ukrywaną treść, wtedy proces ukrywania wiadomości może zostać przedstawiony następującym wzorem:

$$y_1 = E(x, m).$$

Zaproponowana zmiana polega na przekształceniu histogramu obrazu w taki sposób, aby zatracone zostały własności zachodzące pomiędzy sąsiadującymi pikselami. Osiągnięcie tego celu jest możliwe poprzez ograniczenie zbioru wartości histogramu danej barwy. Funkcję kompresji histogramu oznaczymy jako f^*, funkcja dekompresji jest oznaczona jako f. Proces wbudowywania wiadomości będzie

wymagał najpierw wykonania procesu kompresji histogramu obrazu x przy użyciu funkcji f^*, następnie właściwego wbudowania wiadomości funkcją E i w ostatnim kroku ponownej dekompresji histogramu funkcją f. Uzyskany w ten sposób obraz może być wyrażony przez następujący wzór:

$$y_2 = f\left(E(f^*(x), m)\right).$$

Odczytanie wiadomości wymaga ponownego wykonania na otrzymanym obrazie funkcji kompresji f, tak aby wartości histogramu powróciły do zbioru używanego przez algorytm wbudowujący. W związku z tym istotne jest, aby funkcja dekompresji f^* była funkcją odwrotną do funkcji f:

$$f^*\left(f(z)\right) = z.$$

Ze względu na dyskretną charakterystykę reprezentacji obrazów cyfrowych funkcja f nie jest funkcją odwrotną dla funkcji f^* — zazwyczaj $f\left(f^*(z)\right)$ jest różne od z. To nie powoduje żadnych problemów, dopóki zmiany wprowadzane do histogramu przez kompresję i dekompresję obrazu nie są zauważalne.

Aby nie wprowadzać dodatkowych zniekształceń do obrazu, przedstawione funkcje f i f^* powinny także być monotoniczne — powinny być rosnące lub malejące:

$$\forall x_1, x_2 : x_1 > x_2 \Rightarrow f(x_1) \geq f(x_2) \wedge f^*(x_1) \geq f^*(x_2),$$
$$\forall x_1, x_2 : x_1 > x_2 \Rightarrow f(x_1) \leq f(x_2) \wedge f^*(x_1) \leq f^*(x_2).$$

Na potrzeby niniejszej pracy przyjęto następujące postacie funkcji kompresji i dekompresji histogramu:

$$f(x) = x + \left\lfloor \frac{x}{a} \right\rfloor, \quad f^*(x) = x - \left\lfloor \frac{x}{a+1} \right\rfloor,$$

gdzie: x oznacza wartość piksela, a to stała (współczynnik kompresji), $\lfloor x \rfloor$ oznacza funkcję podłoga — największą liczbę całkowitą nie większą od x.

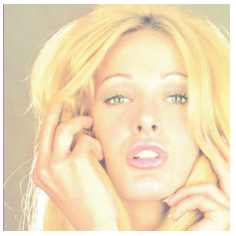
Funkcja $f(x)$ ma zbiór wartości taki sam jak zbiór wszystkich możliwych wartości x — dla obrazów z głębią koloru wynoszącą 8 bitów na każdą składową koloru jest to zbiór liczb całkowitych od 0 do 255. Przykładowo, dla $a = 10$, zbiór wartości funkcji $f^*(x)$ to liczby całkowite od 0 do 232. Wykonanie funkcji dekompresji spowoduje odtworzenie wartości do zbioru 0-255, ale niektóre pary wartości zostaną połączone. Przykładowo, dla $a = 10$ funkcja f^* przekształci wartości 219 i 220 na wartość równą 200. Funkcja f przekształca wartość 200 na 220.

3. Stegoanaliza

W celu zbadania odporności zaproponowanego algorytmu do obrazów pozyskanych z bazy SIPI, każdy o wymiarach 512 × 512 pikseli, wprowadzono 17 476 bajtów danych, co oznacza, że ukryto maksymalną liczbę danych, która może być umieszona przy wykorzystaniu tylko jednego bitu każdego piksela nośnika. Ukrycie danych zostało wykonane przy wykorzystaniu własnej implementacji przedstawionego algorytmu.

Opracowana w środowisku Microsoft Visual Studio 2012 aplikacja została wykonana w języku C#. Aplikacja umożliwia ukrycie danych w plikach o rozszerzeniu bmp, z 24-bitową głębią koloru. Na potrzeby łatwej analizy uzyskanych wyników przyjęto, że ukrywane dane będą miały formę znaków zapisanych zgodnie ze standardem UTF-8.

Wykorzystane w procesie testowania aplikacji i algorytmu obrazy, przedstawione poniżej, będą nazywane odpowiednio: *f16*, *girl*, *house* i *pepper*. Obrazy, do których wiadomość została dodana zgodnie z algorytmem przedstawionym w [5], będą oznaczane jako *nazwa_BCH*, zaś obrazy, w których dane ukryto za pomocą zaproponowanego algorytmu, jako *nazwa_mBCH*.

Rys. 1, 2, 3, 4. Obrazy f16, girl, house i pepper. Źródło: Baza SIPI: http://sipi.usc.edu/database/database.php?volume=misc&image=10#top

3.1. Ataki wizualne

Wprowadzanie dużej liczby zmian do nośnika prowadzi do powstania artefaktów w obrazie. Jeżeli dane są wprowadzane sekwencyjnie, ewentualnie w sposób pseudolosowy, a nośnik posiada obszary jednolitych barw, bądź też nasycenie kolorów będzie maksymalne lub minimalne, to w takich miejscach mogą pojawić się widoczne artefakty. Naturalnie, mogą one pozostać niezauważone w całym obrazie, jednak podczas analizy tylko odpowiedniej grupy bitów obrazu, co do których istnieje podejrzenie wprowadzenia informacji, zmiany te będą łatwo rozpoznawalne. Największą wartość ataki wizualne mają w stosunku do obrazów, które są zbudowane w oparciu o ustaloną paletę barw (np. GIF). Zmiana LSB indeksu barwy zazwyczaj wiąże się z wprowadzeniem istotnych zmian do nośnika.

Należy jednak zaznaczyć, że w przypadku poprawnie dobranego nośnika i wykorzystania algorytmu wprowadzającego ograniczoną liczbę zmian do nośnika, ten typ ataków nie ma zastosowania.

Ze względu na zastosowanie kodów BCH do zmniejszenia liczby wprowadzanych modyfikacji, stegoanaliza wizualna nie ma zastosowania do proponowanego algorytmu. Na obrazach poniżej znajduje się porównanie najmniej znaczących bitów obrazu oryginalnego i obrazu z wbudowaną wiadomością. Jak można zauważyć, brakuje jakichkolwiek widocznych oznak wbudowania wiadomości.

Rys. 5. LSB obrazu girl. Źródło: opracowanie własne

Rys. 6. LSB obrazu girl z wbudowaną wiadomością. Źródło: opracowanie własne

3.2. Atak chi-kwadrat

Idea ataku została przedstawiona w [1]. W niniejszej pracy zostanie ona przybliżona. Głównym celem ataków statystycznych jest porównanie teoretycznej oczekiwanej dystrybucji częstotliwości steganogramów z pewną dystrybucją, która występuje w obserwowanym nośniku.

Najistotniejszym problemem jest uzyskanie oczekiwanej dystrybucji częstotliwości (np. częstotliwości występowania pewnego zjawiska po wbudowaniu wiadomości). Częstotliwość ta nie może zostać określona na podstawie badanego materiału, ponieważ mógł on zostać zmieniony w wyniku wprowadzania steganogramu. W zdecydowanej większości przypadków stegoanalityk nie posiada dostępu do oryginalnego nośnika, w związku z tym pozyskanie z niego wspomnianej częstotliwości bądź porównanie z nią wartości otrzymanej z podejrzanego nośnika nie jest możliwe.

W oryginalnym nośniku za teoretyczną oczekiwaną częstotliwość możemy przyjąć średnią arytmetyczną dwóch częstotliwości danej pary wartości. Dla algorytmu LSB zmianie ulegają najmniej znaczące bity, więc operacja ta nie zmienia wartości sumy tych dwóch częstotliwości. Liczba nieparzystych wartości częstotliwości jest

przenoszona na liczbę wartości parzystych i odwrotnie. Ponieważ suma pozostaje stała, średnia arytmetyczna jest taka sama dla danej pary wartości, zarówno w nośniku oryginalnym, jak i przenoszącym ukrytą treść. Właściwość ta pozwala na wyznaczenie teoretycznej oczekiwanej dystrybucji częstotliwości z losowej próbki. Dzięki temu oryginalny nośnik nie jest wymagany do przeprowadzenia ataku.

Stopień podobieństwa dystrybucji w obserwowanej próbce i dystrybucji teoretycznej jest miarą prawdopodobieństwa, że proces wbudowania wiadomości został wykonany. Stopień podobieństwa jest wyznaczany przy wykorzystaniu testu chi-kwadrat. Podstawą testu jest przypisanie pewnych obserwacji do kategorii. Jest to wykonywane w następujących krokach:

1. Zakładamy, że istnieje k kategorii i posiadamy jedynie losowe próbki do obserwacji. Każda obserwacja musi wpadać do jednej i tylko jednej kategorii. Kategorie oznaczają indeksy palety barw, gdzie kolory mają parzyste indeksy w posortowanej palecie. W celu zachowania ogólności skupiamy się na nieparzystych wartościach par wartości atakowanego obrazu. Ich minimalna teoretyczna oczekiwana częstotliwości musi być większa niż 4. W celu zachowania tego warunku można ujednolicić kategorie.
2. Teoretyczna oczekiwana częstotliwość w kategorii i po wbudowaniu jednolicie rozmieszczonej wiadomości jest równa:

$$n_i^* = \frac{\left|\{kolor \middle| \, posortowanyIndeksKoloru \in \{2i, 2i+1\}\}\right|}{2}.$$

3. Zmierzona częstotliwość występowania w wybranej losowej próbce jest równa:

$$n_i = \left|\{kolor | posortowanyIndeksKoloru = 2i\}\right|.$$

4. Statystyka χ^2 jest dana jako $\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - n_i^*)^2}{n_i^*}$ z $k-1$ stopniami swobody.

5. p oznacza prawdopodobieństwo naszej statystyki, pod warunkiem że rozkłady n_i i n_i^* są równe. Powyższe jest obliczane poprzez całkowanie funkcji gęstości:

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma(\frac{k-1}{2})} \int_0^{\chi^2_{k-1}} e^{-\frac{x}{2}} x^{\frac{k-1}{2}-1} dx.$$

W ramach niniejszego opracowania do obrazów testowych *f16*, *girl*, *house* i *pepper* wprowadzono 17 576 bajtów danych, co odpowiada pełnej pojemności nośnika dla przedstawionego algorytmu. Wyznaczenie prawdopodobieństwa zostało wykonane

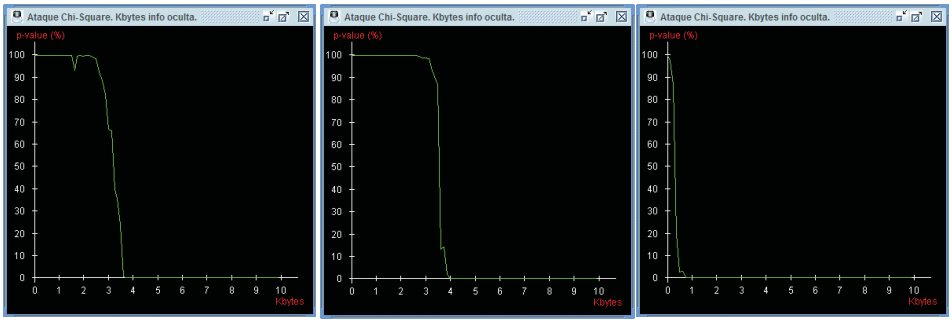
z wykorzystaniem oprogramowania StegSecret [6]. Poniższa tabela przedstawia porównanie oszacowania długości wbudowanej wiadomości dla obrazów oryginalnych z wiadomością wbudowaną algorytmem LSB z wykorzystaniem kodu BCH [5] oraz przedstawionym w niniejszej pracy algorytmem zmodyfikowanym, oznaczanym jako mBCH.

Tabela 1

Oszacowanie długości wbudowanej wiadomości

	Obraz oryginalny [kB]	BCH [kB]	mBCH [kB]
f16	0	0	0
girl	0	0	0
house	0	0	0
pepper	3,1	3,6	0,2

W powyższym zestawieniu zauważalne jest, że dla obrazów *f16*, *girl* oraz *house* atak chi-kwadrat nie był w stanie wykryć żadnych zmian wprowadzonych do obrazu. Dla obrazu *pepper* rozmiar oszacowanej wiadomości dla nośnika oryginalnego i z wiadomością wbudowaną przy wykorzystaniu algorytmu BCH są porównywalne. W przypadku algorytmu mBCH wartość znacząco spadła. Poniższe obrazy prezentują wykres prawdopodobieństwa długości wiadomości dla obrazu *pepper*.



Rys. 7, 8, 9. Prawdopodobieństwo wbudowania wiadomości o określonej długości dla obrazów pepper, pepper_BCH, pepper_mBCH. Źródło: opracowanie własne

3.3. Atak RS

Pomiar właściwości statystycznych najmniej znaczących bitów (LSB) obrazu jest zadaniem trudnym i nie zawsze daje oczekiwane efekty (patrz 3.2). Pomimo że LSB wydają się być losowe, to jednak można odnaleźć pewne związki pomiędzy nimi a bitami bardziej znaczącymi. Wykorzystanie tych zależności zostało

zaproponowane przez J. Fridrich w [2] i [3] i określone mianem stegoanalizy RS dla obrazów kolorowych, jak i tych w odcieniach szarości. Technika ta przegląda obraz w poszukiwaniu grup pikseli regularnych (R) i pojedynczych (S), bazując na pewnych ich właściwościach. Następnie przy wykorzystaniu odpowiednich częstotliwości tych grup w danym obrazie, obrazie otrzymanym z oryginalnego ze zmienionymi LSB oraz w obrazie uzyskanym poprzez ulosowienie LSB oryginalnego obrazu, stara się przewidzieć liczbę wprowadzonych zmian. Szczegółowy opis ataku może zostać znaleziony w [2] i [3].

W celu weryfikacji przyjętych założeń dokonano analizy obrazów oryginalnych z wiadomością wbudowaną przy użyciu algorytmu LSB wykorzystującego kod BCH oraz algorytmu zmodyfikowanego (mBCH). Testy zostały przeprowadzone z wykorzystaniem oprogramowania StegSecret [6]. Wiadomość umieszczono tylko w pikselach koloru niebieskiego, dlatego też dane w poniższym zestawieniu będą oznaczały procent pikseli koloru niebieskiego, które mogły zostać zmodyfikowane.

Tabela 2

Wyniki analizy algorytmem RS

	Obraz oryginalny	BCH	mBCH
f16	1,11%	31,99%	7,30%
girl	0,87%	31,91%	2,71%
house	0,20%	33,30%	7,86%
pepper	8,69%	40,73%	6,14%

Dane zaprezentowane w powyższej tabeli wskazują na znacznie lepsze właściwości algorytmu zmodyfikowanego. W przypadku obrazu *f16* algorytm RS wykazał zmianę 1,11% pikseli barwy niebieskiej, dla obrazu *f16_BCH* 31,99%, zaś dla obrazu *f16_mBCH* 7,30%. Dla obrazu *girl* wartość ta wyniosła 0,87%, *girl_BCH* 31,91%, a *girl_mBCH* 2,71%. Obraz *house* wg algorytmu RS posiadał 0,20% zmienionych pikseli, obraz *house_BCH* 33,30%, *house_mBCH* 7,86%. Dla obrazu *pepper* było to 8,69%, *pepper_BCH* 40,73%, zaś dla *pepper_mBCH* 6,14%.

Przyjmuje się, że próg, dla którego można określić, że pewien obraz przenosi ukrytą treść, jest zawarty w przedziale 5-10% (w zależności od typu obrazu). W analizowanych danych wyraźnie widać, że dla algorytmu BCH każdy z obrazów zostałby oznaczony jako podejrzany. Naturalnie niezbędne byłoby obniżenie liczby danych ukrywanych w obrazie tak, aby nadal pozostały niezauważone. Dla algorytmu zmodyfikowanego wyniki są znacznie lepsze. Dla obrazów *f16* i *house* jest to wartość niewiele większa niż 7%, w związku z tym obraz ten najprawdopodobniej nie zwróci uwagi stegoanalityka. Dla obrazu *girl* wartość ta wynosi zaledwie 2,71%, a dla obrazu *pepper* jest niższa niż dla oryginalnego nośnika.

4. Wnioski

Zaproponowany w niniejszej pracy algorytm, opierający się na propozycji z [5], cechuje się znacznie lepszą odpornością na ataki statystyczne. Jest to zauważalne nie tylko w przypadku stegoanalizy algorytmem RS, lecz także ataku chi-kwadrat. Zmniejszenie artefaktów pozwala na wprowadzanie do nośnika dużo większej liczby danych, niż by to miało miejsce w przypadku algorytmu pierwotnego, jednocześnie pozwalając na skuteczne omijanie wybranych ataków stegoanalitycznych.

Zastosowana w artykule funkcja kompresji histogramu może zostać zastąpiona inną, spełniającą przedstawione wymagania. Możliwe jest także manipulowanie współczynnikiem *a* w trakcie ukrywania wiadomości w danym obrazie tak, aby zapewnić jak największą niewykrywalność przekazu. W powyższym przypadku należy jednak pamiętać o przekazaniu wartości parametru *a* dedykowanemu odbiorcy.

Artykuł wpłynął do redakcji 13.08.2014 r. Zweryfikowaną wersję po recenzji otrzymano 9.10.2014 r.

Niniejsza praca jest współfinansowana ze środków NCBiR na lata 2013-2018 w ramach projektu „ROTOR".

LITERATURA

[1] Westfeld A., Pfitzmann A., *Attacks on Steganographic Systems*, Third International Workshop, IH'99, Dresden, Germany, September 29 October 1, 1999 Proceedings, 2000, 61-76.

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

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

[4] Marcal A.R.S., Pereira P.R., A *Steganographic Method for Digital Images Robust to RS Steganalysis*, Second International Conference, ICIAR 2005, Toronto, Canada, September 28-30, 2005, Proceedings.

[5] Kaczyński K., *Steganografia z wykorzystaniem cyklicznych kodów korekcji błędów*, Biul. WAT, 62, 4, 2013, 267-277.

[6] http://stegsecret.sourceforge.net/ dostęp 11.07.2014.

K. KACZYŃSKI

Reducing vulnerability of modified LSB algorithm to a chosen statistic attacks

Abstract. The LSB algorithm is one of the most studied steganographic algorithms. There are several types of attacks that can detect the fact of conducting cover communication — chi-square attack and RS. This paper presents modification of the LSB algorithm which introduces fewer changes to carrier than the original LSB algorithm. Modified algorithms use a compression function, which significantly hinders the detection process. This paper also includes a description of main steganalytic methods along with their application to the proposed modification of the LSB algorithm.

Keywords: steganography, cyclic code, error correction codes, LSB, BCH, chi-square, steganalysis