POLITECHNIKA BIAŁOSTOCKA

WYDZIAŁ INFORMATYKI

KATEDRA Systemów Komputerowych

PRACA DYPLOMOWA MAGISTERSKA

TEMAT: Analiza algorytmów ukrywania w dźwięku

WYKONAWCA: Marcin Tomaszewski
Imię i nazwisko

PODPIS: .....

PROMOTOR: dr inż. Eugenia Busłowska
Imię i nazwisko

BIAŁYSTOK 2006 r.

Spis treści

Wstęp

Celem niniejszej pracy jest zaprezentowanie istniejących technik i metod steganograficznych ze szczególnym naciskiem na algorytmy wykorzystujące jako kontener dźwięk cyfrowy.

W pracy zostały przeprowadzone analizy porównawcze dwóch wybranych algorytmów steganograficznych, których celem było określenie która z nich zapewnia wyższy stopień bezpieczeństwa przy pomocy badania amplitudy dźwięku. Przeprowadzone badania miały również na celu sprawdzenie czy styl muzyczny kontenera ma znaczenie dla bezpieczeństwa ukrywanych danych.

1. Wprowadzenie do steganografii

Steganografia jest nauką, której celem jest opracowywanie mechanizmów ukrywania danych przed niepowołanymi oczami. Najlepszą definicję wysnuł Duncan Sellars w "An Introduction To Steganography":

"The goal of steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second secret present" [1]

Celem steganografii jest ukrycie wiadomości wewnątrz innej wiadomości w taki sposób, aby przeciwnik nawet nie był w stanie wykryć, że występuje druga, sekretna informacja.

Problem steganograficzny został precyzyjnie określony przez Simmonsa na przykładzie "problemu więźniów".

Alice i Bob siedzą w więzieniu, w oddzielnych celach, bez możliwości prywatnej rozmowy. Aby uzgodnić plan ucieczki muszą wykorzystać otwarty kanał komunikacyjny w postaci listów, które podlegają cenzurze. Cenzor Wendy nie przepuści żadnej wiadomości zaszyfrowanej lub też wydającej się niebezpieczną. W przypadku jednego, silnie monitorowanego kanału komunikacyjnego klasyczna kryptografia jest nieskuteczna, wiadomość musi być tak przemycona, aby sprawne oko Wendy nie zauważyło nic niebezpiecznego. Tu do gry wchodzi steganografia, umożliwiająca więźniom swobodną wymianę informacji.[2]

1.1. Rys historyczny

Termin steganografia pochodzi z języka greckiego. Dosłownie znaczy "ukryte pismo" od słów steganos - ukryty i graphos - pismo.

Historia ukrywania informacji sięga dalekich czasów przed Chrystusem, pierwszy udokumentowany przypadek użycia technik steganograficznych pojawia

się w pismach Herodota (484 - 425 p. n. e.), który opisuje historię Histiausa uwięzionego przez perskiego władcę Dariusa. Herodot dokładnie opisuje sposób w jaki Histiaus przekazał informację o swoim położeniu do swego szwagra. Wiadomość została ukryta pod włosami niewolnika, który został wysłany z nic nie znaczącym i nie wzbudzającym jakichkolwiek podejrzeń listem ze strony straży Dariusa. Niewolnikowi zgolono włosy, w następnej kolejności na skórze głowy Histiaus wytatuował mu prawdziwą wiadmość. Gdy włosy odrosły, niewolnik bez trudu przeszedł przez perskie straże, które nie wykryły żadnych tajnych informacji w przewożonym przez niewolnika liście.[1]

Na przestrzeni wielu lat techniki steganograficzne rozprzestrzeniły się znacznie, wzrosła różnorodność stosowanych metod tajnego przekazywania informacji. Do tego celu wykorzystywane były np. tabliczki drewniane pokryte warstwą wosku służące do nauki pisania. Informacja była grawerowana na drewnie a następnie pokrywana równą warstwą wosku, doskonale wygładzającą powierzchnię deszczółki i skutecznie przykrywającą wiadomość. Zdarzały się przypadki zwinięcia listu w kulkę i pokrycie jej woskiem, która w następnej kolejności połykana była przez posłańca.

W miarę upływu czasu, kiedy sztuka pisania stała się bardziej powszechna, a jako materiały były wykorzystywane już papier i inkaust, techniki steganograficzne ewoluowały. Pojawiły się specjalnie spreparowane substancje, zwane atramentami symptaycznymi, którymi można było napisać wiadomość w sposób niewidoczny. W dalekich Chinach lub Egipcie popularny stał się sok z cytryny. Napis wykonany sokiem z cytryny po wyschnięciu całkowicie znikał. Jako atramenty sympatyczne wykorzystywano również mleko jak i w niektórych przypadkach urynę. Wiadomości napisane powyższymi substancjami odczytywano po uprzednim podgrzaniu papieru. Wysoka temperatura utleniała węgiel zawarty w substancjach organicznych użytych jako atrament, przez co litery ciemniały i stawały się widoczne. Organiczne atramenty sympatyczne miały dwie duże wady - były łatwe do wykrycia oraz istniało duże ryzyko uszkodzenia papieru podczas podgrzewania materiału. Znacznie bardziej zaawansowane techniki wykorzystywały substancje nieorganiczne. W ok. 250 roku przed Chrystusem Filon z Bizancjum zastosował kwas garbarski do napisania tajnej wiadomości. Podgrzewanie materiału nic nie

dawało, informacja nadal pozostawała ukryta przed nieodpowiednimi osobami. Aby ją odczytać, należało papier pokryć solami żelaza, które wchodziły w reakcję chemiczną z kwasem, przez co postawione znaki ciemniały uwidaczniając pismo. Niestety technika ta też była niedoskonała - czas pomiędzy napisaniem wiadomości a jej odczytaniem musiał być stosunkowo krótki.

Pierwszą pracą opisującą metody ukrywania wiadomości była Steganographica napisana w 1665 roku przez Gaspari Schotti, który oparł ją na podstawie badań niemieckiego mnicha Johannesa Trithemiusa. Praktycznie każde kolejne opracowanie odwoływało się do Steganographici uzupełniając zawarte w niej wiadomości. Powstała w 1883 roku Cryptographie militaire przedstawia wiele reguł, będących podwalinami dzisiejszej steganografii. Napisana 24 lata później w 1907 roku Les Filigranes Charlesa Biqueta przez wiele lat była podstawą znaków wodnych.

Największy rozwój steganografii nastąpił w czasie Drugiej Wojny Światowej, zwłaszcza ze strony niemieckiego wywiadu. Przykładem może być przechwycona przez wywiad amerykański pozornie bezpieczna depesza:

Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils.[3]

Czytając jedynie drugą literę każdego słowa otrzymano:

Pershing sails form NY June 1.[3]

Metoda zapisywania kluczowych dla wywiadu informacji polegająca na odczycie poszczególnych znaków wiadomości była wielokrotnie wykorzystywana. Dopiero w momencie wynalezienia systemu mikrokropek technika ta poszła do lamusa. Wg. J. Edgara Hoovera, dyrektora FBI było to "arcydzieło wrogiego wywiadu". Technika ta została zdradzona przez byłego agenta niemieckiego aparatu szpiegowskiego, gdyby nie ten fakt, prawdopodobnie amerykanie nigdy nie dowiedzieliby się o tym systemie.

Mikrokropki były to zdjęcia wykonane w bardzo wysokiej rozdzielczości miniaturyzowane do rozmiarów zwykłej kropki umieszczanej np. w liście lub znaczkach pocztowych. Wyselekcjonowanie mikrokropki i jej wielokrotne powiększenie pozwalało odczytać tajny przekaz. Technika ta pozwalała na przesłanie bardzo dużej ilości danych, łącznie z mapami, szkicami lub planami, będącymi prawdziwym rarytasem dla wrogiego wywiadu.

W czasie trwającej wojny obowiązująca cenzura starała się ograniczyć nie tylko wydawane publikacje, ale również wszelkiego typu rebusy czy krzyżówki, będące doskonałym medium dla technik ukrywania wiadomości. W dobie II Wojny Światowej, gdy w USA prasa codzienna docierała do niektórych obszarów kraju z opóźnieniem nawet tygodniowym, bardzo popularne stało się wysyłanie dzienników lub tygodników rodzinie zamieszkałej w rejonach o trudnej dostępności do prasy. Otworzyło to kolejne możliwości dla ukrywania informacji, zwłaszcza, że poczta była wielokrotnie poddawana cenzurze. System ten zwany szyfrem książkowym polegał na dziurkowaniu cienką igłą poszczególnych liter w słowach zawartych w gazetach. Adresat wiadomości odczytywał wypunktowane litery składając całe słowa i zdania.

Steganografia analogowa jednak oddała pole na rzecz steganografii cyfrowej, która powstała w momencie upowszechnienia się liczących maszyn cyfrowych. Powstały nowe, ogromne możliwości, nowe mechanizmy niezwykle trudne do wykrycia bez przeprowadzenia szczegółowych a zarazem czasochłonnych analiz.

1.2. Idea ukrywania

Idea ukrywania jest jedna i podstawowa – potencjalny agresor nie może dowiedzieć się o istnieniu tajnej wiadomości. Kryptografia nie dysponuje narzędziami umożliwiającymi to – wiadomość mimo że bezpiecznie zaszyfrowana jest widoczna – a to już wystarczająco duże zagrożenie.

W przypadku opisanego przez Herodota zdarzenia steganografia była jedynym narzędziem umożliwiającym przekazanie wiadomości. Tylko i wyłącznie wiadomość niewidoczna mogła zostać przepuszczona przez straże Dariusa.

Czasami jawne istnienie informacji, mimo że bezpiecznie zaszyfrowanej już jest zagrożeniem dla niej. Po przechwyceniu zaszyfrowanych informacji można je

zniszczyć w sposób nieodwracalny uniemożliwiając dostarczenie ich do odbiorcy. Mimo faktu, że szyfr nie został złamany transmisja została przerwana i wymiana danych nie doszła do skutku. Każda wiadomość zaszyfrowana wzbudza podejrzenia – informacja musi być naprawdę ważna skoro została zamknięta przed niepowołanymi oczami, informacja skutecznie ukryta, która nie jest widoczna nie wzbudzi żadnego zainteresowania. Zdjęcia z wakacji, zdjęcia swojego psa lub kota nie zainteresują nikogo poza ich właścicielem, dlatego też umieszczając w nich dane można zapewnić im bardzo wysoki poziom bezpieczeństwa faktem, że nie wzbudzają zainteresowania osób trzecich.

Kolejną cechą mechanizmów steganograficznych jest możliwość „znakowania" plików graficznych lub multimedialnych w celu zaznaczenia np. praw autorskich. Wkomponowanie tzw. Znaku wodnego (ang. Watermark) do sygnału kontenera w taki sposób, aby niemożliwe było jego usunięcie bez znacznej utraty jakości zapewnia ochronę praw autorskich. Jest to wykorzystywane min. W systemach zarządzania prawami DRM (ang. Digital Rights Management).

1.3. Systematyka steganografii

Tajna wiadomość może zostać osadzona praktycznie w każdym typie sygnału. Jednakże w zależności od formatu sygnału stosowane są różne techniki jej umieszczania. Wiele mediów charakteryzuje się różnymi cechami, więc nie można sprecyzować jednego – uniwersalnego algorytmu pasującego do każdego medium. Różne cechy sygnału przykrywającego wpływają na dobór nie tylko algorytmu, ale również na wybór miejsca osadzenia danych. Ze względu na tę cechę steganografię komputerową można podzielić na sześć głównych klas:

- substitution systems – zastępowanie nieistotnych elementów kontenera poszczególnymi tajnymi danymi.
- transform domain techniques – osadzenie informacji w zmodyfikowanej przestrzeni sygnału, np. w dziedzinie częstotliwościowej
- spread spectrum techniques – rozszerzanie widma sygnału – często wykorzystywane w dźwiękach

- statistical methods - ukrywanie informacji poprzez zmodyfikowanie z góry określonych cech statystycznych kontenera.
- disortion techniques – zanieczyszczanie sygnału poprzez dodanie białego szumu zawierającego tajne informacje
- cover generation methods – osadzania wiadomości w momencie tworzenia kontenera. Sygnał przykrywający jest specjalnie tworzony w celach steganograficznych.[4]

Aby zastosowanie technik steganograficznych zapewniało wysoki poziom bezpieczeństwa należy odpowiednio przygotować sygnał będący kontenerem wiadomości. Użycie nieodpowiedniego sygnału może nie zapewnić bezpieczeństwa, a wręcz narazić dane na wykrycie. Przykładem źle wybranego medium może być doskonale biały obraz – umieszczenie w nim nawet najdoskonalszą techniką porcji danych spowoduje zachwianie doskonałości bieli i modyfikacje mogą być widoczne nawet gołym okiem – taka sytuacja jest niedopuszczalna. Siła steganografii nie leży jednak jedynie po stronie sygnału przykrywającego. Bezpieczny stegosystem musi spełnić następujące warunki:

- kontener nie może zostać zmodyfikowany w sposób na tyle znaczny, że zmiany będą widoczne okiem nieuzbrojonym.
- struktura kontenera musi pozostać nienaruszona przy jednoczesnym upakowaniu danych zewnętrznych
- osadzona informacja musi być odporna na uszkodzenia lub celowe przetwarzanie sygnału poprzez edycję, samplowanie, kompresję stratną
- ukryta wiadomość (a przynajmniej jej fragment) musi być możliwa do wyodrębnienia nawet w przypadku utraty części kontenera.[1]

Każdy stegosystem narażony jest na uszkodzenia jak i celowe ataki ze strony potencjalnych agresorów. Przykładem niecelowego uszkodzenia osadzonej informacji jest potraktowanie obrazu posiadającego ukryty przekaz osadzony prostą techniką LSB dowolnym edytorem graficznym – np. zastosowanie opcji sharpen, ang. Wyostrzanie. Jest to jeden z częściej wykorzystywanych filtrów, zwłaszcza przy obróbce grafiki dopiero co zmniejszonej. Przy prostych technikach

ukrywających może to skutecznie zniszczyć tajną informację.

Do ataku na stegosystem może dojść w dwóch przypadkach. W pierwszym główną rolę zagrał przypadek – atakujący sprawdzając dużą ilość danych przypadkiem natrafił na kontener z danymi. Jest to przypadek bardzo rzadko spotykany i raczej nieprawdopodobny. Drugi przypadek jest bardziej realny. Atakujący miał podstawy sądzić, że przechwycony sygnał zawiera „coś" ukrytego, np niepokojące szumy i zakłócenia w sygnale dźwiękowym. Jeżeli to będzie miało miejsce wówczas stegosystem można uznać już za bliski kompromitacji. Całkowita kompromitacja systemu steganograficznego nastąpi wówczas, gdy atakujący system skutecznie wyodrębni dane. Klasyczny atak na stegosystem przebiega trzyetapowo:

- wykrycie (ang. Detection) – atakujący może znaleźć się w trzech stanach na etapie wykrywanie osadzonej informacji:
    - może wykryć osadzoną informację – oznacza to pełną kompromitację systemu
    - może nie wykryć informacji popełniając błąd II typu – wówczas stegosystem nie zostaje skompromitowany i osadzone informacje pozostają bezpieczne
    - może niepoprawnie wykryć osadzoną informację popełniając błąd I typu – stegosystem jest na granicy kompromitacji, jednak informacja jest nadal bezpieczna.[4]

  Konstruując algorytm steganograficzny dąży się do osiągnięcia prawdopodobieństwa wystąpienia błędu II typu równego 1.

  Prawdopodobieństwo wykrycia zależy zarówno od użytej techniki osadzania danych jak i od jakości i typu medium przykrywającego. Jeżeli zostanie wybrany nieodpowiedni sygnał, taki w którym łatwo jest zauważyć „zanieczyszczenia" wprowadzone przez funkcję steganograficzną to prawdopodobieństwo wykrycia znacznie wzrasta. Dysponując pewną bazą wiedzy o statystyce danego typu sygnału można przeprowadzić atak statystyczny, przynoszący dobre efekty. Udowodniono również, że niektóre aplikacje steganograficzne osadzając informację narażają ją na łatwe wykrycie przez atakującego. Aplikacje te działają w sposób

szablonowy i znając sposób jej działania wiadomo gdzie i czego szukać – niektóre wręcz zapisują informacje o sobie w pliku wynikowym.

Na etapie wykrywania osadzonej informacji można posłużyć się następującymi metodami ataku:

- stego-only attack – metoda o bardzo niskiej skuteczności ze względu na dużą ilość dostępnych metod stganograficznych. Posiadanie samego stegosystemu nie gwarantuje poprawności wykrycia czegokolwiek
- known cover attack – atakujący ma dostęp zarówno do stegosystemu jak i pierwotnej wersji kontenera nieposiadającego ukrytych danych
- known message attack – atakujący dysponuje pełną treścią ukrytego przekazu – atak mający na celu skompromitowanie algorytmu.
- Chosen stego attack – algorytm jest znany dla agresora – żadna wiadomość nie jest bezpieczna, prawdopodobieństwo skompromitowania równe 1.
- chosen message attack – atak polegający na poddawaniu próbki wcześniej przygotowanych informacji działaniu różnych algorytmów steganograficznych w celu wykrycia podobieństwa lub wzorca do znanej metody. Technika ataku przypominająca metodę łamania haseł brute-force.
- known stego attack – atakujący dysponuje kompletną wiedzą na temat uzytego algorytmum, posiada przechwycony stegosystem jak i oryginalny sygnał użyty jako medium przykrywające.[4]

- Wyodrębnianie (ang. Extraction) – jeżeli na etapie wykrywania nie został popełniony błąd I typu istnieje wysokie prawdopodobieństwo wyodrębnienia i poprawnego odczytania ukrytych informacji z kontenera.

- Usuwanie (ang. Deletion) – może zdarzyć się, że atakujący po wyodrębnieniu informacji z kontenera będzie starał się zniszczyć przekaz, nie wzbudzając podejrzeń zarówno u nadawcy jak i odbiorcy przekazu, lub informację tą sfałszować. Atakujący może usiłować zniszczyć wiadomość w momencie gdy nie uda mu się poprawnie ich wyodrębnić i odczytać. Przy wykorzystaniu wielu technik zniszczenie ukrytych danych jest procesem stosunkowo prostym i niezbyt czasochłonnym.

W praktyce występują trzy typy algorytmów steganograficznych, których podobieństwo zasady działania można zauważyć do algorytmów kryptograficznych:

- Prosty algorytm steganograficzny (ang. Pure steganography).

Definicja 1:

$\partial =< C, M, D, E >$, gdzie C jest zbiorem możliwych kontenerów (ang. Covers), M jest zbiorem wiadomości, przy czym musi zachodzić własność $|C| \geq |M|$. $E : CxM \rightarrow C$ jest funkcją osadzająca wiadomość w kontenerze, $D : C \rightarrow M$ jest funkcją wyodrębniającą, przy zachowaniu własności: $D(C(m,c)) = m$, gdzie dla każdego $m \in M$ i każdego $c \in C$ nazywamy prostym algorytmem steganograficznym.[4]

Cechą główną prostego algorytmu steganograficznego jest to, że wcześniej, pomiędzy nadawcą i odbiorcą wiadomości nie musi dojść do wymiany lub uzgodnienia klucza – wystarczy znajomość algorytmu przez obie strony. Aby algorytm był bezpieczny musi być utrzymany w ścisłej tajemnicy, ujawnienie jego zasady działania spowoduje totalną kompromitację.

Formalnie osadzanie przebiega następująco:

$E : CxM \rightarrow C$

Gdzie C jest zbiorem wszystkich możliwych kontenerów zaś M zbiorem wiadomości. Proces wyodrębniania to: $D : C \rightarrow M$. Przy czym konieczne jest spełnienie zależności

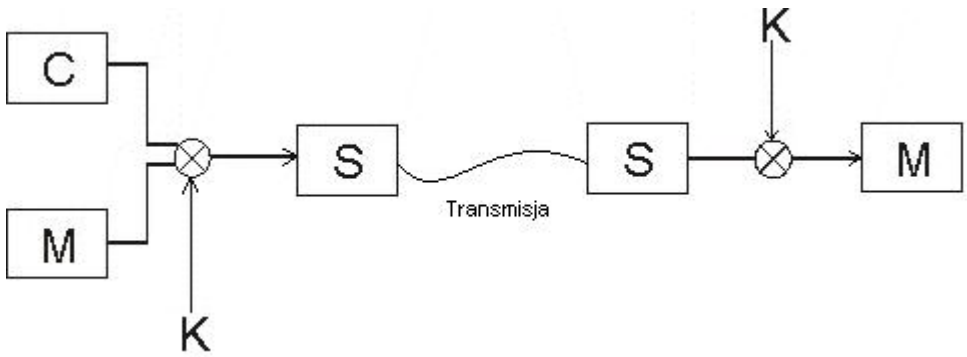
$|C| \geq |M|$

Zaletą prostego algorytmu steganograficznego jest fakt, że przed wysłaniem informacji nie musi nastąpić przekazanie klucza otwartym kanałem informacyjnym, gdzie może być narażony na przechwycenie przez osoby trzecie. Jednakże stałość algorytmu znacząco odbija się na jego bezpieczeństwie. Dysponując odpowiednio dużą ilością danych do analizy można zrekonstruować strukturę algorytmu, jak to ma miejsce przy łamaniu prostych szyfrów podstawieniowych lub przedstawieniowych.

- Algorytm z kluczem prywatnym

Definicja 2:

$\partial =< C, M, K, D_k, E_k >$, gdzie C jest zbiorem możliwych kontenerów, M jest zbiorem tajnych wiadomości, gdzie musi wystąpić własność $|C| \geq |M|$, K jest zbiorem kluczy prywatnych, $E_k : CxMxK \to C$ i $D_k : CxK \to M$ z zachowaniem zależności $D_k(E_k(c,m,k)) = m$ dla każdego $m \in M$, każdego $c \in C$ i każdego $k \in K$ nazywamy algorytmem steganograficznym z kluczem prywatnym.[4]



Rys. 1. Osadzanie i wyodrębnianie z wykorzystaniem klucza prywatnego

Aby uniezależnić się od samego algorytmu, uważając aby nie został złamany lub odtajniony wprowadzono algorytmy steganograficzne bazujące na kluczu prywatnym. Występuje tu pewne podobieństwo do algorytmów kryptograficznych symetrycznych, gdzie istniał jeden klucz – zarówno do szyfrowania jak i deszyfrowania wiadomości.

Warunkiem koniecznym do spełnienia jest wcześniejsza propagacja klucza – zarówno nadawca jak i odbiorca muszą posiadać identyczny. Przesyłanie otwartym i niebezpiecznym kanałem wiąże się z ryzykiem przechwycenia klucza. Aby

zminimalizować te ryzyko można posłużyć się algorytmami kryptograficznymi służącymi do ustalania kluczy na odległość.

Algorytm Diffie-Hellmana:

„publicznie znane są p, która jest wystarczająco dużą liczbą pierwszą, oraz generator alfa dla Zp. Protokół ma następującą postać:

1. Alice wybiera Z_a, Bob wybiera Z_b, gdzie $Z_a, Z_b \leq p-2$. liczby Z_{ai} Z_b są trzymane w tajemnicy.
2. Alice przesyła Bobowi liczbę $Ca = \alpha^{Za} \bmod p$, Bob przesyła Alice liczbę $Cb = \alpha^{Zb} \bmod p$.
3. jako uzgodniony klucz przyjmowana jest liczba $k = \alpha^{Za \cdot Zb} \bmod p$. Alice oblicza k jako $(Cb)^{Za} \bmod p$, Bob korzysta z równości $k = (Ca)^{Zb} \bmod p$

Algorytm Diffie-Hellmana mimo że nie korzysta z żadnego szyfrowania przy przekazywaniu wartości obliczonych przez Alice i Boba uznawany jest za metodę względnie bezpieczną. Agresor podsłuchujący transmisję pomiędzy Alice i Bobem nie jest w stanie obliczyć klucza k mimo znajomości danych publicznych. Aby obliczyć klucz k, atakujący musiałby wyliczyć logarytm dyskretny do wyznaczenia liczb Z_a i Z_b z C_a i C_b, gdzie nie znany jest jeszcze efektywny algorytm wyznaczania logarytmu dyskretnego.[5]

Algorytm Diffie-Hellmana nie jest jedynym algorytmem wymiany klucza. Istnieją jego odmiany, równie dobrze nadające się do ustalenia kluczy steganograficznych takie jak: protokół STS (Stadion-To-Station) lub protokół MTI (Matsumoto Takashima Imai).[5]

Stałe używanie jednego klucza wiąże się z niebezpieczeństwem jego utraty lub odkrycia przez osoby niepowołane. Lepszym rozwiązaniem są klucze jednorazowe - ich bezpieczeństwo jest znacznie wyższe niż jednego, stałego klucza, gdyż nawet w przypadku odkrycia klucza przez osoby trzecie, praktycznie nie istnieje prawdopodobieństwo odszyfrowania danych zakodowanych innymi kluczami. Wadą kluczy jednorazowych jest konieczność posiadania karty kodów lub książki kodowej (jak np. w przypadku Enigmy). W algorytmach steganograficznych istnieje możliwość zastosowania klucza jednorazowego poprzez

klucz sesji.

Klucz sesji jest obliczany z funkcji: $K = H(feature)$ [4]

Gdzie H jest niezmienną, z góry określoną funkcją, natomiast feature jest pewną niezmienną cechą kontenera. Cecha ta musi być tak wybrana, aby osadzanie wiadomości nie zaburzyło jej – w tym przypadku wyodrębnienie informacji stanie się niemożliwe, gdyż nie będzie sposobu na obliczenie klucza sesji przez adresata. Cechą tą może być np długość sygnału dźwiękowego, rozmiar kontenera w bajtach, rozdzielczość użytego obrazu, lub kombinacja tych cech razem.

Zastosowanie klucza sesji zapewnia bardzo wysoki poziom bezpieczeństwa. Praktycznie każdy stegosystem tworzony jest na bazie innego klucza, nie ma potrzeby wcześniejszego ustalania lub wymiany kluczy pomiędzy adresatem i nadawcą. Niezbędne natomiast jest ustalenie funkcji H() i cechy.

- Algorytm z kluczem publicznym

Istnieją pewne sytuacje gdzie użycie jednego klucza do osadzania wiadomości, jak i do wyodrębniania nie zapewnia odpowiedniego poziomu bezpieczeństwa, lub w przypadku gdy rozpropagowanie klucza czy też jego ustalenie jest zbyt niebezpieczne, stegosystem może zostać przedwcześnie skompromitowany.

Zupełnie jak w przypadku kryptografii symetrycznej posiadając wiedzę na temat użytego algorytmu i dysponując odpowiednią ilością danych wejściowych do analizy, atakujący jest w stanie odtworzyć klucz.

W 1990 roku w ten sposób udało się złamać kryptograficzny algorytm symetryczny DES, przez wiele lat uznawany za niełamliwy metodami innymi niż brute-force. Eli Biham i Adi shamir opracowali metodę kryptoanalizy różnicowej bazującej na tym, że znany był algorytm i dysponowano dużą ilością danych wejściowych (w przypadku 16 rundowego DES wymagane były 2^{55} znane teksty jawne) dla złamania klucza. Trzy lata później nastąpił kolejny przełom, Matusi opracował metodę kryptoanalizy liniowej, bazującej na liniowej aproksymacji do „odgadnięcia” klucza.[5]

Użycie jednego klucza do szyfrowania a drugiego klucza do odszyfrowywania dostało miano kryptografii asymetrycznej. W steganografii

asymetrycznej jeden klucz przeznaczony jest do osadzania wiadomości, natomiast drugi do jej wyodrębniania.

Klucz za pomocą którego następuje osadzania informacji w kontenerze nosi miano klucza publicznego będącego dostępnym każdemu. Klucz służący do wyodrębniania danych jest kluczem prywatnym, znanym jedynie dla adresata wiadomości. Parę kluczy publiczny-prywatny należy wygenerować przed rozpoczęciem transmisji.

Definicja 3:

Stegosystem z kluczem publicznym jest potrójnym złożeniem probabilistycznych algorytmów $S = (SG, SE, SD)$, gdzie $SG(1^k)$ generuje parę kluczy $(p_k, s_k) \in PKxSK$.

$SG(1^k) \rightarrow \{p_k, s_k\}$

SE bierze publiczny klucz p_k należący do PK oraz tekst jawny m należący do $\{0,1\}$. Funkcja SE zwraca wiadomość ukrytą s:

$SE(p_k, m) = s$

SD wykorzystując prywatny klucz s_k należący do SK i stegosystem s, zwraca wyodrębnioną wiadomość m.

$SD(s_k, s) = m$ [4]

Standardowo klucz publiczny jest kluczem mocno rozpowszechnionym i znanym szerokiemu kręgu ludzi. Wiadomość osadzona/zaszyfrowana za pomocą klucza publicznego danej osoby może zostać wyodrębniona/odszyfrowana jedynie przez właściciela klucza publicznego, posiadającego również klucz prywatny.

Aby techniki steganograficznej były jeszcze skuteczniejsze i bardziej bezpieczne można posiłkować się algorytmami kryptograficznymi. Przed osadzeniem do kontenera wiadomość jest szyfrowana wybranym algorytmem i dopiero szyfrogram jest umieszczany w sygnale przykrywającym i transmitowany do odbiorcy.

Wcześniejsze użycie kryptografii, oprócz zmniejszenia prawdopodobieństwa prawidłowego odczytania osadzonego przekazu poprzez podwojenie siły systemu, zwiększa również prawdopodobieństwo popełnienia błędu II typu przy ataku na stegosystem. Agresor nie wiedząc o fakcie wykorzystania siły szyfrowania, nawet po prawidłowym wyodrębnieniu danych otrzyma przypadkowo wyglądający ciąg

znaków. Prawdopodobieństwo domyślenia się jakiego algorytmu szyfrującego użyto jest bliskie 0.

1.4. Wybór kontenera

Od prawidłowego wyboru sygnału przeznaczonego na kontener w dużej mierze zależy skuteczność ukrywania danych. Idealny kontener to taki, który został stworzony bezpośrednio w celu ukrycia w nim danych i jest ściśle prywatny dla jego właściciela. Nie istnienie żadnej innej kopii uniemożliwi wykorzystanie ataku typu Known-Cover-Attack. Przygotowanie kontenera zależy od typu i wielkości danych jakie ma przechowywać. Niedopuszczalne są sygnały mało zróżnicowane – takie jak obraz jednolicie kolorowy lub o bardzo niskim kontraście i różnorodności barw, sygnał dźwiękowy posiadający prawie płaską amplitudę, dźwięk stworzony z niewielkiej ilości mało zróżnicowanych ampli.

Wskazane jest, aby sygnał kontenera jak i dane do ukrycia spełniały w jak największym stopniu zależność funkcji podobieństwa (ang. Similarity function)

Definicja 4:

Niech C będzie niepustym zbiorem, Funkcja $sim : C^2 \to (-\infty, 1]$, jest nazywana funkcją podobieństwa dla C, jeżeli dla x,y należącego do C zachodzi:

Sim(x,y)=1 $\Leftrightarrow$ x=y

Dla x<>y sim(x,y)<1

Dla systemów steganograficznych:

$sim(C(E(c,m))$ zbiega do 1, dla każdego $c \in C$ i $m \in M$, gdzie C to zbiór wszystkich możliwych kontenerów, $E(c,m)$ jest funkcją osadzającą wiadomość m w kontenerze c.[4]

Im większe podobieństwo stegosystemu i kontenera tym sygnał kontenera został mniej zmodyfikowany poprzez osadzanie i przeprowadzenie stegoanalizy będzie znacząco utrudnione.

Przy doborze medium należy pamiętać o spełnieniu zależności:

$|C| \geq |M|$

W przeciwnym wypadku wiadomość nie zostanie prawidłowo ukryta lub może ulec zniszczeniu całość lub jej część.

2. Wybrane media

Wraz z rozwojem technologii cyfrowej steganografia musiała ewoluować. Opracowanie cyfrowych nośników informacji stworzyło olbrzymie możliwości dla ukrywania informacji. Dane nie były już reprezentowane analogowo posiadając znacznie mniejsze możliwości, sposoby osadzania jak i wyodrębniania danych uległy pełnej automatyzacji, wymagany nakład pracy zmniejszył się do operacji kilkuminutowych. To co kiedyś wymagało sporej dawki wiedzy, wyobraźni i niemałej ilości czasu zostało zredukowane do automatycznych programów wykonujących wszystkie żmudne czynności za człowieka. Sam sygnał cyfrowy stworzył kolejne możliwości - opracowywane techniki steganograficzne są co raz doskonalsze pod względem bezpieczeństwa i niezmienności kontenera. Proste zasady oszukiwania ludzkich zmysłów zostały wzbogacone o skomplikowane reguły matematyczne pozwalające uzyskać lepsze efekty w stosunku do ery przedkomputerowej. Różnorodność cyfrowych formatów danych stworzyło wręcz nieograniczone możliwości dla steganografów. Każda informacja zapisana cyfrowo, niezależnie od jej formatu nadaje się w mniejszym lub większym stopniu do bycia kontenerem na inną wiadomość, zależnie od jej konstrukcji.

2.1. Tekst

Tekst jawny jest jednym z najstarszych mediów wykorzystywanych w steganografii. Pozornie nieskomplikowana wiadomość zapisana na skrawku papieru lub w formie elektronicznej może nieść pewną dawkę informacji niejawnej.

Podczas II Wojny Światowej steganografia tekstowa wykorzystywała specjalnie preparowane wiadomości, gdzie przy znajomości algorytmu lub klucza można było wyodrębnić inna, niejawną wiadomość. Przykładem historycznym może być zapisywanie wiadomości w ściśle określonych literach każdego słowa.

Tekst jako kontener posiada jednak spore ograniczenia:

- jest w stanie pomieścić niewielką ilość danych
- wykrycie informacji transportowanej w tekście jawnym jest łatwe do wykonania, nawet bez specjalistycznego oprogramowania
- bardzo łatwo uszkodzić lub całkowicie zniszczyć informację w nim zapisaną.

Algorytmy steganograficzne bazujące na kontenerach tekstowych można zakwalifikować do dwóch grup:

1. algorytmy osadzające informację w sposób dający możliwość przechowywanie stegosystemu w formie niecyfrowej (np. Jako wydrukowany arkusz papieru)
2. algorytmy wyłącznie cyfrowe.

Do pierwszej kategorii zalicza się grupę mechanizmów zaproponowanych przez J. T. Brassila określanych jako line-shift-coding, word-shift-coding, feature-coding, metodę semantyczną i syntaktyczną. Do grupy drugiej zalicza się algorytm białych znaków.

2.1.1. Line shift coding

Algorytm line-shift-coding opiera się na precyzyjnym manipulowaniem położenia wiersza tekstu w stosunku do wierszy nad i pod nim. Przesunięcie powinno mieć stałą wartość i wynosi ok. 1/300 cala. Zaletą tego algorytmu jest fakt, że ukryta informacja nie zostanie zniszczona nawet po wydrukowaniu dokumentu. Badania dowiodły, że możliwe jest poprawne wyodrębnienie osadzonej wiadomości nawet z 10 pokolenia oryginału przy fotokopiowaniu dokumentu. Natomiast podstawową wadą jest to, że prawdopodobieństwo wykrycia informacji przez niepowołane osoby jest bardzo wysokie i nie wymaga specjalistycznych narzędzi.
Algorytm zakłada, że przesunięciu ulegają tylko wybrane wiersze tekstu, więc dany wiersz może znaleźć się w jednym z trzech stanów:[6]

$S = \{0, -1, 1\}$

gdzie S=0 – brak przesunięcia wiersza

S=1 – przesunięcie wybranego wiersza w górę

S=-1 – przesunięcie wybranego wiersza w dół.

Aby umożliwić prawidłowe wyodrębnienie danych z tekstu należy założyć, że położenie nie wszystkich wierszy może ulec zmianie. Założono, że wiersze parzyste pozostają w stanie S=0, natomiast jedynie wiersze nieparzyste mogą przyjąć stan S=1 lub S=-1. Nie zachowując tej zasady nie będzie można określić punktu odniesienia do obliczenia przesunięcia. Przykładowo będące w sąsiedztwie wiersze:

wiersz 1 - stan S=-1

wiersz 2 - stan S=-1

wiersz 3 - stan S=-1

są niepoprawnie zakodowane. Niemożliwe jest jasne określenie ich stanu. Równie dobrze każdy z nich może być w stanie S=0 jak i S=1.

Jeżeli jednak wiersz 1 i 3 będą w stanie S=0, wówczas analizując dokument można zauważyć, że wiersz 2 jest na pewno w stanie S=-1.

Line-shift-coding bazuje na słowach kodowych. Słowo kodowe jest zdefiniowanym zbiorem wartości przesunięcia poszczególnych wierszy dokumentu, w ten sposób można oddać reprezentację bitową, np.:

S=-1 – wartość bitowa 1

S=1 – wartość bitowa 0

Oznaczanie słów kodowych jest dowolne i zależne od implementacji algorytmu. Za pomocą 19 wierszy tekstu można określić 2^{19} słów kodowych co daje wartość: 524588 wszystkich możliwych słów kodowych.

Systemy wyodrębniające osadzoną informację mogą opierać się na jednej z dwóch metod:

- baseline detection decision rule

Niech wiersze i-1 oraz i +1 nie będą przemieszczone w pionie (stan S=0), wiersz i będzie w stanie S={-1,1}, niech l będzie odległością pomiędzy wierszami i-1 oraz i, oraz odległością pomiędzy wierszami i oraz i+1 wówczas reguła wykrywania przesunięcia wiersza przyjmuje postać:

jeżeli l(i-1,i)>l(i,i+1) wówczas wiersz jest w stanie S=-1

jeżeli l(i-1,i)<l(i,i+1) wówczas wiersz jest w stanie S=1

jeżeli l(i-1,i)=l(i,i+1) wówczas wiersz jest w stanie S=0[6]

- centroid detection decision rule

Niech S_i-1 oraz S_i będą centroidami pomiędzy wierszami i-1 oraz i, a także wierszami i+1 oraz i w dokumencie kontenera. Niech t_i-1 oraz t_i będą odpowiednimi centroidami pomiędzy wierszami w niezmodyfikowanym dokumencie kontenera, wówczas reguła ma postać:

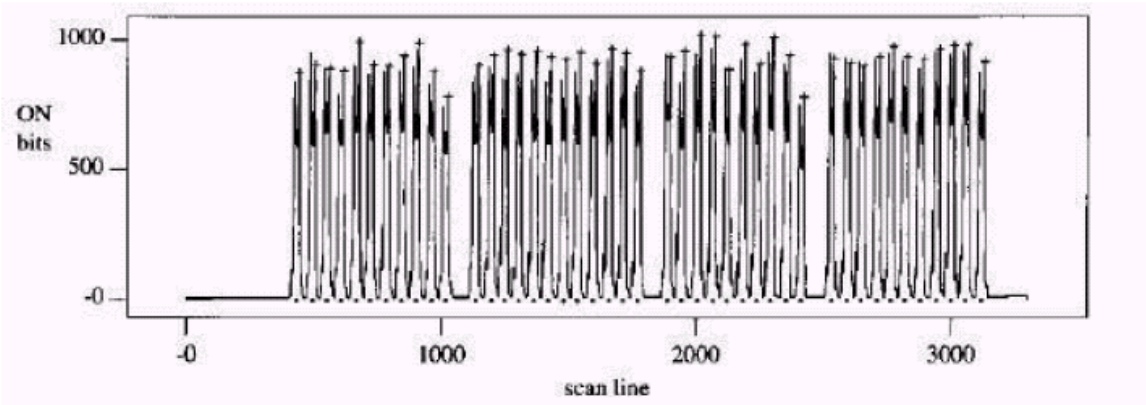
jeżeli s_i-1 – t_i-1 > s_i – t_i wówczas wiersz jest w stanie S=-1

jeżeli s_i-1 – t_i-1 < s_i – t_i, wówczas wiersz jest w stanie S=1

w pozostałym przypadku wiersz będzie w stanie S=0.

Wykorzystując regułę baseline detection decision rule istnieje pewne prawdopodobieństwo nieprawidłowego zinterpretowania stanów badanego wiersza. Stan S=-1 może zostać odczytany jako stan S=1 i odwrotnie, co może spowodować błąd I typu w procesie wykrywania osadzonej informacji. Reguła bazująca na centroidzie jest niepodatna na tego typu błąd.

Przeprowadzając atak na stegosystem oparty na algorytmie line-shift-coding można posłużyć się analizą profilu zależności pomiędzy położeniem wierszy w dokumencie. Posiadając wykres z łatwością można zauważyć odchylenia od normy.



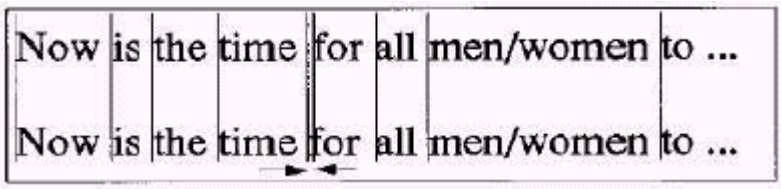
Rys. 2 przykładowy profil zależności pomiędzy wierszami

2.1.2. word-shift-coding

W odróżnieniu od line-shift-coding ta metoda opiera się na manipulowaniu położeniem nie całego wiersza w pionie, ale położeniem pojedynczych słów lub bloków słów w wierszu względem osi X. Najlepiej do tego celu nadają się dokumenty wyrównane do obu krawędzi strony, aczkolwiek gdy nie ma spełnionego tego warunku użycie word-shift-coding jest również możliwe. Sprecyzowano dwie odmiany:

- w każdym wierszu dokumentu odnajdywana jest największa i najmniejsza odległość pomiędzy słowami. Następnie największa odległość jest pomniejszana o stałą wartość, natomiast najmniejsza powiększana o wartość identyczną.
- Wiersz dokumentu dzielony jest na trzy bloki słów. Przesunięciom podlega jedynie blok środkowy względem bloku skrajnie lewego lub

skrajnie prawego.



Rys. 3. przykładowe przesunięcie w lewo słowa „for”

Metoda manipulowania pozycją słów w wierszu również bazuje na słowach kodowych. Słowo kodowe może zatem być zapisane za pomocą 3 stanów.

S=0 gdy słowo nie uległo przesunięciu

S=1 gdy słowo uległo przesunięciu w lewo

S=-1 gdy słowo uległo przesunięciu w prawo

Ilość słów kodowych możliwych do zapisania w danym wierszu dokumentu jest ściśle związana z rodzajem użytego algorytmu oraz długością wiersza – nie w każdym można zmodyfikować położenie takiej samej ilości słów.

Word-shift-coding jest metodą bezpieczniejszą od line-shift-coding. Istnieje mniejsze prawdopodobieństwo wykrycia przesunięcia pojedynczych słów lub bloków słów niż całego wiersza. Jednakże wraz ze wzrostem bezpieczeństwa osadzonej informacji wzrasta ryzyko jej uszkodzenia. Jak w przypadku line-shift-coding możliwe było poprawne wyodrębnienie wiadomości z 10 pokolenia fotokopii dokumentu, tak w przypadku word-shift-coding niemożliwe to już jest w przypadku drugiego pokolenia.

Aby móc odczytać osadzoną informację należy posiadać zarówno stegosystem jak i oryginalny dokument, który został użyty jako kontener. W przeciwnym wypadku nie sposób określić przesunięcie których słów zostało zmienione. Rozszerzając ten klasyczny algorytm o możliwość wykorzystania klucza można wyeliminować konieczność posiadania dokumentu oryginalnego. W tym przypadku klucz może opisywać, które słowa lub bloki słów mogą ulec przesunięciu.[7]

2.1.3. Feature-coding

Najpopularniejszym kodowaniem za pomocą wybranych cech jest wykorzystanie wysokości liter należących do zbioru:

$L = \{b, d, f, h, k, l, t\}$

Algorytm feature-coding w tym przypadku opiera się na zmianie wysokości znaków należących do zbioru L. Słowo kodowe może zostać zapisane jako „wydłużenie" lub „skrócenie" wysokości poszczególnych znaków. Znak mogą znajdować się w stanach[6]

S=0 jeżeli wysokość pozostaje niezmieniona

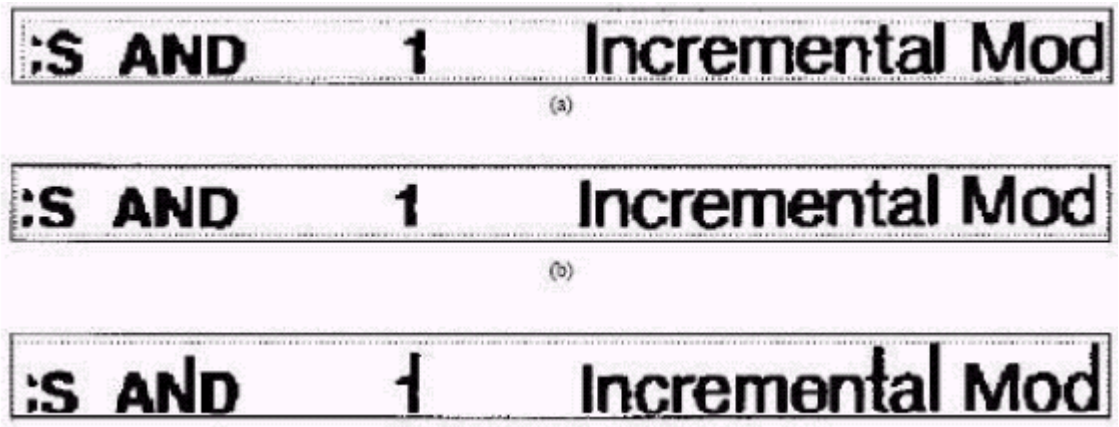
S=1 jeżeli wysokość uległa zmianie.

Dzięki czemu można słowa kodowe zapisywać binarnie.

Proces osadzania danych tą metodą przebiega dwuetapowo:

1. wysokości wszystkich znaków ze zbioru L są zwiększane lub zmniejszane o losową wartość w celu utrudnienia osobie niepowołanej określenia ich wysokości w stanie S=0.
2. Słowo kodowe jest zapisywane poprzez modyfikację wysokości wybranych znaków ze zbioru L poprzez zwiększenie lub zmniejszenie.

Metoda ta zapewnia dość wysoki poziom bezpieczeństwa ukrytych danych, gdyż niewielka zmiana wysokości znaków jest ignorowana przez ludzki mózg. Jednak w celu prawidłowego wyodrębnienia informacji niezbędny jest dokument oryginalny. Użycie klucza steganograficznego może wyeliminować konieczność posiadania niemodyfikowanego oryginału dokumentu – klucz może ściśle wskazywać na znaki w których zapisane jest słowo kodowe.



Rys. 4. Działanie algorytmu feature-coding.

2.1.4. Metoda białych znaków

Jest to jedyna metoda ukrywania danych w tekście jawnym, która nie może być wykorzystywana do dokumentów przeznaczonych do druku. Występowanie białych znaków w tekście jest widoczne dla człowieka jedynie w tedy, gdy są one umieszczone pomiędzy znakami lub słowami oraz gdy występują na początku wiersza. Białe znaki ulokowane na końcu wiersza są dla człowieka nie widoczne – wyjątkiem jest dokument, w którym tekst jest wyrównany do obu marginesów – w tym przypadku nie jest zalecane używanie tej metody.

Poprzez dodanie określonej liczby białych znaków takich jak spacja na koniec wiersza możliwe jest osadzenie słowa kodowego binarnie. Ilość spacji określa maksymalną ilość bitów możliwą do osadzenia w jednym wierszu:
2 spacje reprezentują 1 bit, gdzie 1 spacja oznacza 0 binarne, 2 spacje 1 binarne
4 spacje reprezentują 2 bity
8 spacji reprezentuje 3 bity.[1]

Wiersz 1 __
Wiersz 2 _
Wiersz 3 __

Zakładając, że znak „_” reprezentuje spację, na powyższych 3 wierszach zakodowano binarnie 101.

Im więcej znaków spacji zostanie użyta, tym łatwiej jest wykryć, lub uszkodzić skonstruowany w ten sposób stegosystem.

Możliwa do osadzenia ilość danych jest mocno ograniczona, zakładając kodowanie tylko 2 znakami spacji na wiersz, w dokumencie złożonym z 80 wierszy można osadzić jedynie 10 bajtów danych:

$$I = \frac{il\_wierszy}{8}$$

Zamiast znaków spacji można używać dowolnych innych białych znaków, niewidocznych dla człowieka.

Przez swoją prostotę mechanizm ukrywania danych posiada bardzo duże ograniczenia:

1. jest łatwy do wykrycia
2. bardzo duża podatność na uszkodzenia
3. niewielka możliwa ilość danych do ukrycia
4. wielkość kontenera znacznie wzrasta

2.1.5. metoda syntaktyczna

Metoda syntaktyczna bazuje na ludzkiej tendencji do popełniania błędów ortograficznych, gramatycznych i interpunkcyjnych.

Zakładając, że obie konstrukcje są gramatycznie poprawne:

1) chleb, masło i mleko
2) chleb, masło, i mleko

można założyć, iż w dokumencie oryginalnym, zawsze używana jest forma 1), w której nie występuje przecinek przed spójnikiem „i". Konstruując stegosystem opartym na interpunkcji zakłada się, że:

binarne 0 określone jest jako brak przecinka przed spójnikiem (konstrukcja 1))

binarne 1 określane jest jako występowanie przecinka przed spójnikiem (konstrukcja 2)).[8]

Zapisując w kontener słowo kodowe należy zmodyfikować ustawienie przecinka przed spójnikiem „i" w odpowiednich miejscach tekstu kontenera.

Pewną odmianą tej metody syntaktycznej jest technika bazująca na błędach. Jeżeli poprawną konstrukcją gramatyczną jest występowanie znaku przecinka przed słowem „że" i przypiszemy jej wartość binarną 1, to usuwając znak „," popełniając przy tym błąd gramatyczny można zapisać stan binarny 0.

Obie odmiany algorytmu steganograficznego opartego na syntaktyce charakteryzują się dużym poziomem bezpieczeństwa. W procesie detekcji istnieje bardzo wysokie prawdopodobieństwo popełnienia błędu II typu. Jeżeli użyty jest pierwszy wariant, który bazuje jedynie na poprawnie gramatycznych konstrukcjach wykorzystanie programów do automatycznej korekty dokumentu nie uszkodzi osadzonej zawartości.

Problemem jest dobór kontenera, gdyż również w tym przypadku osadzenie dużej ilości danych jest niemożliwe. Dokument kontenera może zostać specjalnie stworzony w celu osadzenia w nim danych, lub też, korzystając z gotowego tekstu

należy zapewnić jego wystarczającą dużą objętość.

Stegosystemy utworzone metodą syntaktyczną doskonale nadają się do przekazywania ich w formie zarówno drukowanej jak i elektronicznej. Poprzez wydrukowanie ich na papier nie ma możliwości uszkodzenia informacji niejawnej, jak ma to miejsce w technikach line-shift-coding, word-shift-coding, feature-coding lub też metody białych znaków.

2.2. Obrazy graficzne

W erze szeroko rozpowszechnionej komputeryzacji obrazy graficzne stały się łakomym kąskiem dla steganografów. Główną cechą stegosystemów opartych na grafice jest bardzo duża ilość danych, jaką można ukryć, oraz wysoki poziom ich bezpieczeństwa.

Ze względu na budowę formatu graficznego, można je podzielić na dwie główne kategorie:

- obrazy bez kompresji stratnej – są to formaty takie jak standardowy BMP opracowany pierwotnie dla systemu IBM OS/2, a następnie zaadoptowany na praktycznie wszystkie pozostałe platformy. Charakteryzuje się tym, że przechowuje obrazy jedynie jako kombinację 3 składowych barw:

1. czerwony (ang. Red)
2. zielony (ang. Green)
3. niebieski (ang. Blue)

posiada bardzo prostą konstrukcję i składa się z nagłówka, opcjonalnie z palety kolorów i danych obrazu, który jest zapisywany od dołu do góry, jako lustrzane odbicie w pionie. W przypadku obrazów zapisanych w większej ilości kolorów niż 256 paleta kolorów nie występuje, a każdy kolor liczony jest wg

$$K = R + 256 \bullet G + 65535 \bullet B$$

i każdy piksel obrazu reprezentowany jest przez 3 bajty odpowiedzialne za poszczególne składowe koloru. W przypadku obrazów w trybie 256 kolorów, każdy piksel reprezentowany jest przez jeden bajt zawierający indeks do palety kolorów.[9]

- obrazy z kompresją stratną – najpopularniejszym formatem jest jpeg. Standard został opublikowany w 1991 roku i definiuje od 4 typy kompresji:
- sekwencyjny, oparty na DCT
- progresywny, oparty na DCT
- sekwencyjny, bez stratny
- hierarchiczny

kompresja stratna znacząco wpływa na jakość obrazu, dlatego też, format ten jest zalecany do stosowania w przypadku grafiki nie zawierającej dużej ilości ostrych krawędzi i małych detali, np. zdjęcia pejzaży, portrety.

Kompresja stratna to metoda zmniejszania ilości bitów potrzebnych do wyrażenia danej informacji, która nie daje gwarancji, że odtworzona informacja będzie identyczna z oryginałem. Dla niektórych informacji algorytm kompresji stratnej może odtworzyć informację w sposób identyczny.[9]

Kompresja stratna wykorzystuje fakt, że ludzki zmysł wzroku eliminuje najmniej istotne elementy obrazu w celu znacznego zmniejszenia rozmiaru pliku.

W zależności od używanego formatu pliku kontenera istnieją różne metody steganograficzne.

2.2.1. Algorytm LSB

Jedna z najprostszych technik steganograficznych, przeznaczona do ukrywania informacji niejawnej w obrazach bez kompresji stratnej, nieposiadających palety kolorów – czyli np. Plików zapisanych w standardzie BMP TrueColor 24bits. Algorytm LSB bazuje na modyfikacji najmłodszych bitów określonych bajtów, LSB – najmniej znaczący bit (ang. Least Significant Bit).

Jeżeli obraz zapisany jest z 24 bitową głębią, wówczas każdy piksel reprezentowany jest przez 3 bajty, opisujące poszczególne składowe barwy. Do zapisania jednego bajta ukrywanych danych niezbędne są zatem 3 piksele = 9 bajtów.

Kolor biały reprezentowany jako funkcja RGB(255,255,255) w zapisie binarnym

będzie miał postać:

R G B

11111111 11111111 11111111

Poddając modyfikacji najmłodszy bit każdego bajtu uzyska się kolor biały o jeden stopień jasności ciemniejszy

R G B

11111110 11111110 11111110

czyli funkcję RGB(254,254,254). Dla niedoskonałego oka ludzkiego taka zmiana koloru jest niezauważalna i całkowicie ignorowana. Dzięki tej zależności można swobodnie modyfikować najmłodsze bity w obrazie zapisując w nich informację niejawną bez narażania obrazu na znaczny spadek jakości, lub widoczne gołym okiem zmiany.

Przykładowo dysponując obrazem, którego fragment ma postać binarną:

00110110 11101001 01011000 11000011 10101000 00011010 00101011 11001011

i znakiem „K" o reprezentacji ASCII 1001011, można go zapisać w obrazie w następujący sposób:

00110111 11101000 01011000 11000011 10101000 00011011 00101011 11001011

Do zakodowania litery „K" niezbędne było zmodyfikowanie jedynie 3 bajtów oryginalnego obrazu, co nie wpłynie znacząco na dany fragment. W przypadku maksymalnie pesymistycznym wszystkie 8 bajtów wymagałyby zamiany.

Ilość danych możliwych do osadzenia metodą LSB w pliku bmp TrueColor można wyliczyć ze wzoru:

$$\frac{(x \bullet y) \bullet \frac{K}{8}}{8} \qquad \{1\}$$

gdzie X jest szerokością obrazu w pikselach

Y jest wysokością obrazu w pikselach

K jest głębią koloru w bitach.

Oznacza to, że w niewielkim obrazku o rozdzielczości 100px / 100px TrueColor można osadzić 3750 bajtów danych, nie obniżając znacząco jakości obrazu kontenera.[4]

Aby zwiększyć ilość danych możliwych do ukrycia wykorzystano słabość

ludzkich zmysłów do rozpoznawania zmian poszczególnych barw. Zmysł wzroku jest najmniej podatny na zmiany koloru niebieskiego, natomiast najbardziej podatny na zmiany barwy zielonej. Stosunek czułości dla poszczególnych barw ma postać 3:6:1 dla modelu RGB. Oznacza to, że modyfikacja większej ilości młodszych bitów dla barwy niebieskiej nie wpłynie na jakość obrazu.

Dysponując fragmentem obrazu:

00110110 11101001 01011000 11000011 10101000 00011010 00101011 11001011

oraz znakiem ASCII „K" w postaci 01001011, można zapisać go jedynie na 3 bajtach obrazu – nie na ośmiu jak w poprzednim przypadku:

00101101 11010110 00101011

Umożliwia to zapisanie większej ilości danych w tym samym obrazie, jednak kosztem jego jakości.

Proces ukrywania informacji metodą LSB przebiega wg. Schematu:

for i=1 to l(c) do

si<-ci endfor

for i=1 to l(m) do

compute index ji where to storage message bit

sji<-cji endfor

natomiast proces wyodrębniania przyjmuje postać:

for i=1 to l(M) do

compute index ji where the ith message bit is stored

mi<-LSB(cji)

Endor[4]

W procesie wyodrębniania może pojawić się problem, że wraz z poprawnie odczytaną informacją niejawną zostaną wyciągnięte szumy. Wynika to z tego, że standardowo algorytm nie ma informacji o długości ukrytych danych. Aby uniknąć zaśmiecania wyniku wyodrębniania zalecane jest zapisanie przed danymi informację i ich długości na kilku pierwszych bajtach kontenera.

Konstruując system steganograficzny oparty na metodzie LSB należy dokładnie wybrać plik kontenera, jaki ma zostać użyty. W pierwszej kolejności

musi być spełniona zależność:

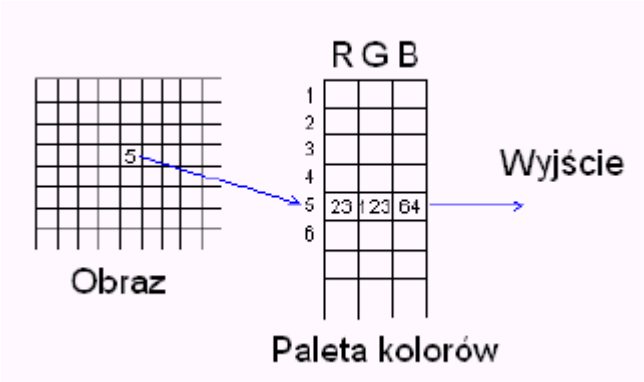
$$|C| \geq |M|$$

gdzie C jest maksymalną pojemnością kontenera w bajtach liczoną ze wzoru {1}, a M jest wielkością osadzanej wiadomości w bajtach.

W zależności od używanego wariantu metody LSB należy odpowiednio dobrać obraz. W przypadku 1 metody, praktycznie nadaje się każdy obraz spełniający zależność $|C| \geq |M|$. w przypadku 2 wariantu istotne jest, aby w obrazie nie występowały płynne przejścia między odcieniami jednej barwy (tzw. Gradient), gdyż modyfikacja najmłodszych bitów może znacznie zaburzyć płynność przejścia.

2.2.2. Obrazy z paletą kolorów

W plikach graficznych, w których występuje paleta kolorów zastosowanie metody zmiany najmłodszego bitu jest niemożliwe. Kolor poszczególnego piksela nie jest reprezentowany przez 3 bajty lecz jedynie 1 bajtem, zawierającym indeks do tablicy – palety kolorów. Zmodyfikowanie najmłodszego bajtu w indeksie spowoduje znaczne zniszczenie obrazu, gdyż indeksy zaczną wskazywać na zupełnie inne kolory.[10]



Rys.5. Paleta kolorów – indeksy

Proces ukrywania danych w obrazach z paletą przebiega następująco:

1. zredukowanie palety barw.

   Proces ten obniża jakość kontenera w stosunku do oryginału poprzez zmniejszenie ilości występujących w nim kolorów. Wykonuje się to poprzez zmianę odwołań indeksów, tak, aby barwy zbliżone do siebie zastępowany były tylko jednym kolorem. Redukcja palety wykonywana jest o wartość

$n = k^2$, gdzie k przyjmuje wartości od 1.

zredukowanie palety 256 kolorów o n=3 spowoduje, że obraz będzie reprezentowany jedynie przez 32 kolory co spowoduje znaczną utratę jakości.

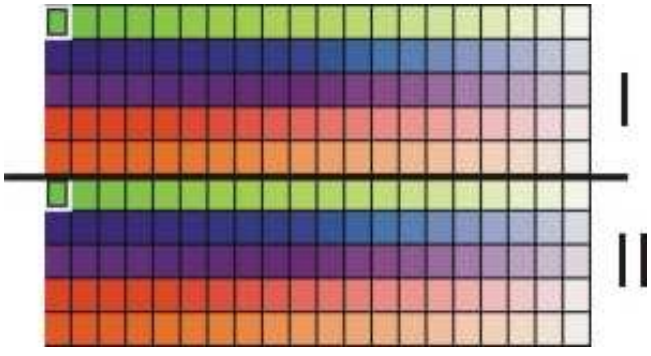
Redukcja ma na celu zapewnienie określonej nadmiarowości danych w palecie, która następnie będzie użyta do osadzania danych.

Najprostszym sposobem na zredukowanie palety jest zastosowanie metody histogramowej, gdzie redukowane są kolory najrzadziej występujące w obrazie – takie do których indeksu jest najmniej odwołań. Jest to metoda zawodna, gdyż niektóre obrazy charakteryzują się tym, że kolor, który nie jest często wykorzystywany może mieć znaczący wpływ na jakość obrazu. Doskonalszą metodą jest redukcja kolorów podobnych:[11]

niech w palecie występują barwy o zapisie dziesiętnym 145 144 i 147, wówczas redukcja kolorów podobnych może wyeliminować wskazania indeksów do kolorów 145 i 147 zastępując je kolorem 144. Spowoduje to spadek jakości obrazu, jednak przy odpowiednim skonstruowaniu kontenera, strata może być mało zauważalna.

2. Zwielokrotnienie palety kolorów.

   Po zredukowaniu ilości kolorów w palecie barw, tablicę należy zwielokrotnić w celu uzyskania identycznego rozmiaru w stosunku do oryginału. Jeżeli paleta zredukowana była ze współczynnikiem n=2 to zwielokrotnienie również musi być przeprowadzone n-razy. Wynikiem zwielokrotnienia jest n identycznych palet umieszczonych jedna pod drugą, przez co dany kolor występuje n-razy.



Rys. 6. Zwielokrotnione palety barw

3. Zmiana indeksów.

   W związku z tym, że część kolorów została usunięta z obrazu, kontener należy przeindeksować w taki sposób, aby „martwe" indeksy odnosiły się do istniejących kolorów w palecie. Wykonuje się to poprzez obliczenie normy luminancji dla poszczególnych barw i wskazanie na indeks najbliższy w palecie. Normę luminancji można wyliczyć z normy euklidesowej:

   $$|W| = \sqrt{((L_{11} - L_{12})^2 + (L_{21} - L_{22})^2 + (L_{31} - L_{32})^2)} \qquad \{2\}$$

   natomiast luminancję jako sumę wartości składowych RGB z uwzględnieniem wag:

   $L = L_1 + L_2 + L_3$, gdzie:

   L1=3R;

   L2=6G;

   L3=1B;

   współczynniki równania {2} L11, L12, i L13 są współczynnikami dla pierwszej porównywalnej barwy, natomiast L12, L22 i L32 są współczynnikami dla drugiej porównywalnej barwy.

4. Właściwe osadzenie danych

   Informacja niejawna kodowana jest na indeksach odwołujących się do palety. Jeżeli tablica kolorów została zredukowana ze współczynnikiem n=2, wówczas jeden kolor występuje 2 razy w palecie. Modyfikując indeksy w kontenerze, tak aby poszczególne piksele odwoływały się zarówno do podstawowej palety jak i zarówno do jej kopii można zakodować, że odwołanie do pierwszej palety jest równoznaczne z binarną wartością 1, natomiast odwołanie do drugiej palety jest równoznaczne z binarną wartością 0. Wówczas osiem pikseli obrazu może przechowywać 1 bajt danych informacji niejawnej. Redukując paletę z wyższym współczynnikiem n możliwe jest zakodowanie 1 bajta niejawnych danych na mniejszej ilości bajtów kontenera.

5. Wymieszanie palety

   W ostatnim kroku, każdą kopię palety należy wymieszać, tak aby wizualnie nie było widać, że występuje kilka identycznych kopii palety, zmieniając

równocześnie odwołania indeksów.

Siłą algorytmu jest fakt, że modyfikując kolory w obrazie poprzez np. Ściemnianie lub rozjaśnianie poszczególnych fragmentów obrazu nie spowoduje utraty osadzonych informacji. Zmianie ulegną jedynie poszczególne kolory w palecie kolorów, jednak indeksy do nich pozostaną niezmienione, przez co nie nastąpi utrata danych. W procesie detekcji istnieje wysokie prawdopodobieństwo popełnienia błędu zarówno I jak i II typu. Stego-only attack jest praktycznie niemożliwy do poprawnego przeprowadzenia – niezbędne jest posiadanie oryginalnego, niemodyfikowanego pliku z obrazem.[10]

2.2.3. DCT

DCT (ang. Discreet Cosinus Transform) – dyskretna transformata cosinusowa to jedna z najpopularniejszych blokowych transformat danych. Jest szczególnie popularna w stratnej kompresji danych[9]

Transformata DCT jest wykorzystywana min. w obrazach formatu jpeg, które dzięki swojej popularności w sieci web stały się jednym z głównych zainteresowań steganografów. Algorytm kompresji stratnej w formacie jpeg przebiega następująco:

1. konwersja obrazu z kanałów RGB na luminancję i 2 kanały chrominancji
2. wstępne odrzucenie części pikseli ze względu na niższą rozdzielczość barwy oka ludzkiego niż rozdzielczości jasności
3. kanały są dzielone na bloki 8x8
4. na wyszczególnionych blokach dokonywana jest dyskretna transformata cosinusowa, zamieniając wartości poszczególnych pikseli na średnią wartość wewnątrz bloku oraz częstotliwości zmian.
5. Zastąpienie średnich wartości wewnątrz bloku przez różnice wobec wartości poprzedniej
6. zamiana wartości zmiennoprzecinkowych na wartości całkowite powodując utratę części danych – kwantyzacja
7. współczynniki DCT umieszcza się tak, aby wartości zerowe występowały

obok siebie

8. współczynniki niezerowe są kompresowane algorytmem Huffmana.

Po dokonaniu kompresji obrazu za pomocą DCT można zauważyć pewne efekty blokowe mogące znacznie wpływać na jakość obrazu, w zależności od użytego stopnia kompresji:

- Obraz niekompresowany o rozmiarze 196662 bajtów:

rys. 7. Płynne przejścia między barwami spirali

- Obraz potraktowany bardzo silną kompresją DCT o rozmiarze 1741 bajtów:

Rys. 8. Widoczne efekty blokowe

W pierwszym etapie działania algorytmu steganograficznego bazującego na plikach w formacie jpeg następuje podział kontenera na bloki o rozmiarze 8x8 pikseli w identyczny sposób co przy dyskretnej transformacie cosinusowej. Zakłada się, że każdy z bloków może przechowywać jedynie 1 bit informacji niejawnej, poprzez zmodyfikowanie np. współczynników kwantyzacji luminancji.[9]

(u,v)	0	1	2	3	4	5	6	7
0	16	11	10	16	24	40	51	61
1	12	12	14	19	26	58	60	55
2	14	13	16	24	40	57	69	56
3	14	17	22	29	51	87	80	62
4	18	22	37	56	68	109	103	77
5	24	35	55	64	81	104	113	92
6	49	64	78	87	103	121	120	101
7	72	92	95	98	112	100	103	99

Rys. 9. Tablica kwantyzacji luminancji wykorzystywana w formacie jpeg.[4]

Osadzanie danych powinno przebiegać, w odróżnieniu od metody LSB lub modyfikacji palety kolorów w najbardziej istotnych elementach obrazu – fakt ten zapewnia wysoki poziom bezpieczeństwa i odporności na uszkodzenia poprzez np. ponowną kompresję, danych niejawnych.

Każdy wybrany blok b_i może przechowywać tylko jeden i-ty bit osadzanych danych. Aby prawidłowo ukryć, a następnie wyodrębnić dane na początku należy ustalić położenie dwóch modyfikowanych współczynników kwantyzacji (u1,v1) i (u2,v2).

Jeżeli $B_i(u_1, v_1) > B_i(u_2, v_2)$ wówczas blok kontenera przechowuje binarną wartość 1, w przeciwnym wypadku 0.

funkcja B_i ma postać:

$Bi = D(b\mathrm{i})$ [12]

Odpowiednio manipulując współczynnikami kwantyzacji dla poszczególnych bloków danych można ukryć dużą ilość informacji w sposób bardzo bezpieczny – dane osadzone są w ważnych punktach obrazu, oraz są odporne na stratną kompresję obrazu.

2.3. Protokoły sieciowe

Powstanie protokołów sieciowych z rodziny stosu TCP/IP stworzyło nowe możliwości dla steganografii cyfrowej. Duża ilość danych przemierzających sieć zarówno globalną jak i lokalną umożliwiła transmitowanie równie dużej ilości informacji niejawnych w bardzo bezpieczny sposób, bez konieczności używania jakiegokolwiek pliku zewnętrznego jako kontenera na dane. Same protokoły posiadające budowę blokową umożliwiają osadzenie pewnej ilości danych.

TCP/IP jest protokołem otwartym, co oznacza możliwość komunikacji między dowolną kombinacją urządzeń, bez względu na ich fizyczną różnorodność. {wiki}

Model stosu protokołów TCP/IP posiada czterowarstwową strukturę:

1. warstwa łącza, obsługująca transmisję pakietów
2. warstwa sieciowa
3. warstwa transportowa, odpowiedzialna za dostarczenie pakietów
4. warstwa aplikacji [13]

Każda transmisja danych oparta o protokoły TCP/IP bazuje na datagramach, strumień danych przeznaczony do wysłania jest dzielony na sprecyzowane na poziomie specyfikacji protokołu paczki danych, opatrzone odpowiednim nagłówkiem zawierające informacje niezbędne do prawidłowego dostarczenia i odebrania ich przez adresata. Algorytmy steganograficzne wykorzystujące TCP/IP opierają się w większości na występowaniu pewnej nadmiarowości danych w nagłówkach pakietów, dzięki czemu sam datagram nie ulega uszkodzeniom lub zmianie. Bardzo duża ilość transmitowanych pakietów w jednostce czasu umożliwia przesłanie dużej porcji danych niejawnych, bez konieczności przygotowywania specjalnego pliku kontenera, co jest zaletą takich systemów. Wadą jest to, że aby skutecznie przesłać osadzone steganograficznie dane do adresata musi nastąpić bezpośrednia transmisja danych pomiędzy nadawcą jak i odbiorcą. Niemożliwe jest przechowywanie sygnału w formie np. pliku, gdyż wszystkie niejawne dane ukrywane są w nagłówkach, które ulegają utracie zaraz po ich odebraniu lub zakończeniu transmisji.

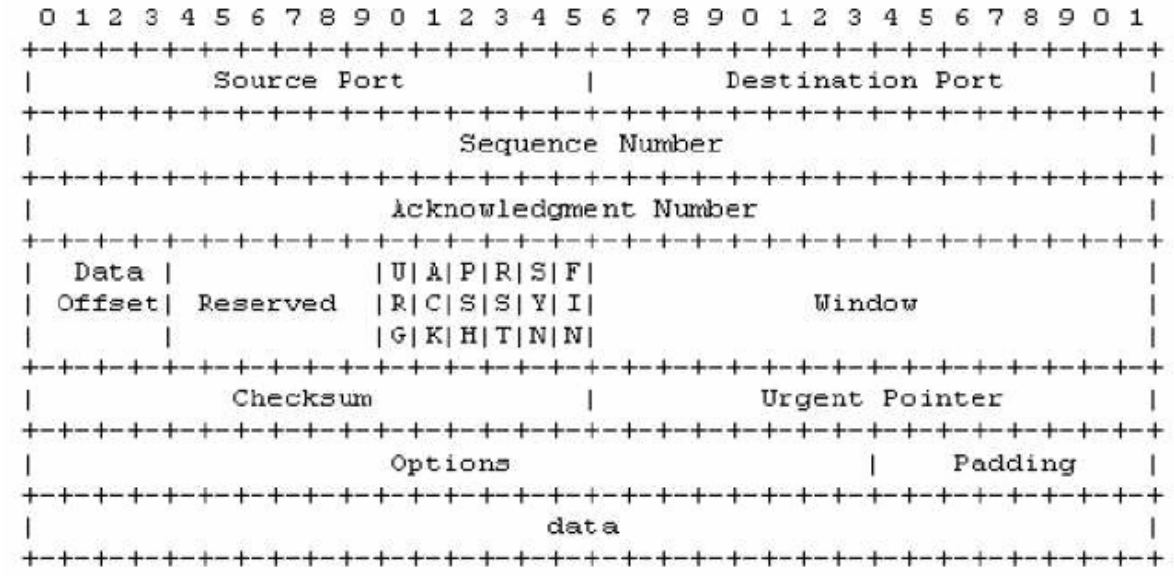
2.3.1. Protokół TCP

Jest to strumieniowy protokół obsługi transmisji pomiędzy dwoma hostami w sieci. Zapewnia wiarygodny sposób dostarczenia pełnego strumienia danych do odbiorcy poprzez kontrolę błędów oraz numery sekwencyjne, dzięki którym możliwe jest ponowne poprawne złożenie danych w całość.

Połączenie TCP następuje w trzech krokach (ang. three-way handshake):

1. Nadawca wysyła pakiet z ustawioną flagą SYN.
2. Jeżeli odbiorca chce odebrać połączenie odsyła pakiet z ustawionymi flagami SYN i ACK.
3. Nadawca wysyła pierwszy pakiet danych z aktywną flagą ACK i nieaktywną już flagą SYN.

Jeżeli host odbiorcy nie może obsłużyć połączenia odsyła pakiet z ustawioną flagą RST. W momencie zamknięcia połączenia wysyłany jest datagram z ustawioną flagą FIN.



Rys. 10. Schemat budowy nagłówka TCP[13]

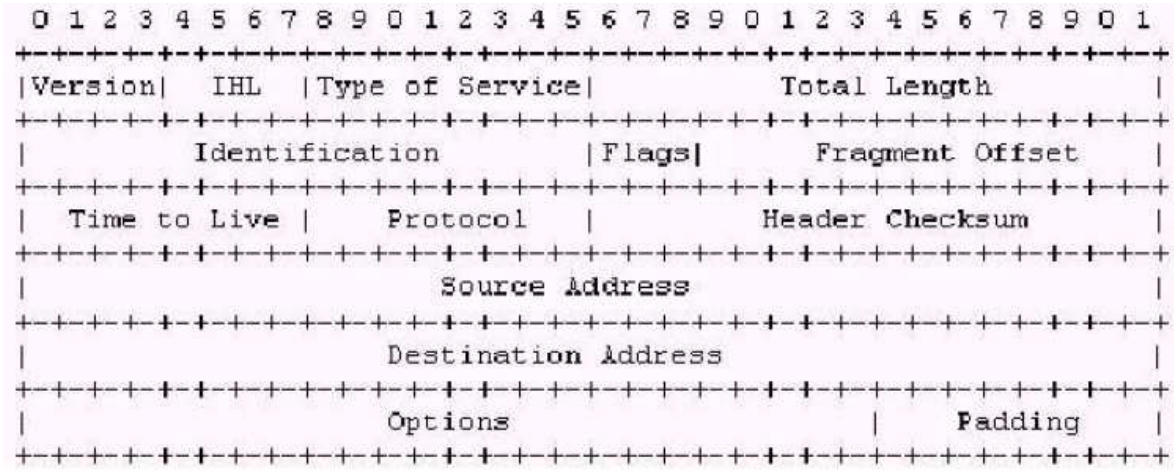
Technika steganograficzna ukrywająca dane w nagłówkach TCP bazuje na 6 bitach przeznaczonych na flagi, odpowiedzialne za odpowiedni interpretację pakietu przez transmitujące hosty. Każda z flag może przyjąć wartości z zbioru {0,1}. istnieją 64 kombinacje ustawienia flag, jednak w trakcie normalnej transmisji danych wykorzystywanych jest jedynie 29 kombinacji.

Ustawienie flagi URG na wartość 0 spowoduje, że odbiorca będzie ignorował

dane zapisane w polu nagłówka Urgent Pointer, co umożliwia na zapisanie tam 16 bitów danych, nie mających żadnego wpływu na poprawność komunikacji danych. Odpowiednio skonstruowany program pozwoli na odczytanie pola Urgent Pointer i złożenie z nich osadzoną wiadomość. [14]

2.3.2. Protokół Ipv4

Protokół odpowiedzialny za przekazanie datagramów pomiędzy dwoma hostami w sieci. Rozpoznawanie i adresowanie hostów polega na identyfikacji po unikatowych w obrębie danej sieci adresach IP. Cechą charakterystyczną Ipv4 jest fragmentacja datagramów wynikająca z różnorodnej konstrukcji sieci, dane pofragmentowane zostają ponownie prawidłowo złożone, pomimo faktu, że mogą zostać odebrane w różnej kolejności, niekoniecznie takiej w jakiej zostały wysłane. Wiąże się to z ustanowieniem łącza wirtualnego przebiegającego poprzez różne węzły sieciowe, tak, że datagramy mogą być transmitowane różnymi drogami.



Rys. 11. Budowa nagłówka protokołu IP

Konstrukcja nagłówka protokołu IP zakłada, że występują w nim trzy bity przeznaczone na flagi wymuszające między innymi proces fragmentacji pakietu:

1. pierwszy bit zarezerwowany
2. drugi bit – flaga DF – jeżeli DF=1 to następuje wymuszenie fragmentacji
3. trzeci bit – flaga MF

Zakładając stałą i niezmienną wartość MTU sieci, czyli maksymalną wielkość datagramu (ang. Maximum Transfer Unit) można manipulować bitem DF w taki sposób, aby możliwe było przesłanie informacji niejawnej bit po bicie w każdej

ramce IP. Możliwe jest to jednak, wyłącznie w sieciach lokalnych o stałej wartości MTU. W sieci globalnej Internet, metoda ta nie sprawdza się ze względu na mogącą wystąpić różnorodność wartości MTU na poszczególnych węzłach transmisyjnych.

Do przesyłania przekazów steganograficznych poprzez sieć Internet można zastosować inną własność protokołu IP. Pole opcji jest zazwyczaj niewykorzystywane w sieci globalnej, sprawia to, że do osadzenia danych można przeznaczyć 5 32 bitowych słów danych. Modyfikując pola version, internet header length oraz identification można przygotować strukturę nagłówka odpowiednią dla osadzenia danych.

Proces osadzania danych przebiega następująco:

1. dwa czterobitowe pola version i internet header length przyjmują odpowiednie wartości po usunięciu pola opcji. Niech bity te będą oznaczone jako [h1...hs]. Stanowią one pierwsze 8 bitów pierwszego słowa nagłówka.
2. Pole identification stanowi pierwsze 16 bitów drugiego słowa nagłówka i niech będą oznaczone jako [i1...is]. Pierwsze 8 bitów pierwszego słowa i drugiego słowa należy traktować indywidualnie. Pierwsze 8 bitów powinno dać wartości [h1...hs]=01000101, natomiast 8 bitów drugiego słowa wartości [c1...cs] mogące zawierać osadzane dane.
3. Pierwsze 8 bitów pierwszego i drugiego słowa powinny być poddane funkcji XOR
4. pozostałe 8 bitów pola identification może zostać wygenerowana losowo i zostać połączone z pierwszymi 8 bitami w celu uzyskania unikatowej wartości, niezbędnej do prawidłowej transmisji danych.[14]

Utworzony w ten sposób datagram, zawierający ukryty przekaz może być transmitowany w sieci Internet. Jeżeli nastąpi konieczność pofragmentowania danych, to przy ponownym jej złożeniu nie nastąpi uszkodzenie zawartych informacji.

Protokoły TCP oraz IP nie są jedynymi protokołami możliwymi do użycia w steganografii, praktycznie dla każdego protokołu możliwe jest opracowanie odpowiedniego algorytmu, jednakże ich wykorzystanie może być ograniczone.

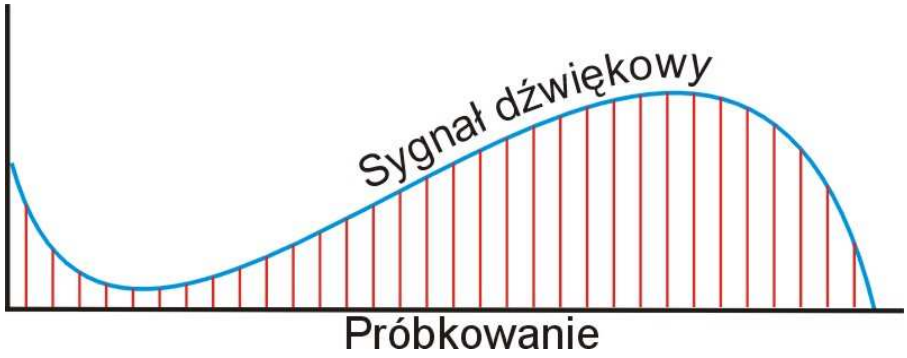
Metody steganograficzne oparte na transmisji sieciowej posiadają wysoki poziom bezpieczeństwa, jednak wykrycie ich jest możliwe poprzez długotrwałe

analizowanie danego fragmentu sieci pod kątem budowy nagłówków. Każda modyfikacja nagłówka może zostać odebrana jako anomalia, jednak momencie gdy powtarza się ona cyklicznie i tylko pomiędzy określonymi hostami w sieci, istnieje wysokie prawdopodobieństwo, że zostanie to odebrane jako celowe działania steganograficzne.

Wadą protokołów sieciowych jako kontenera jest fakt, że niemożliwe jest przechowywanie zabezpieczonych steganograficznie informacji. Możliwa jest jedynie ich transmisja w czasie rzeczywistym.

3. Dźwięk jako kontener

Dźwięk może występować w dwóch postaciach – jako sygnał analogowy oraz sygnał cyfrowy. Aby sygnał dźwiękowy mógł być przetwarzany przez komputer musi on ulec zmianie z analogowego na cyfrowy. Wykonywane jest to poprzez próbkowanie sygnału analogowego i zapisywanie wartości próbek do pliku. Zsamplowany dźwięk traci jednak na jakości w procesie próbkowania, gdyż analizowanie sygnału odbywa się z pewną szybkością, która nie jest w stanie wiernie odwzorować sygnału analogowego.



Rys. 12. Samplowanie dźwięku analogowego

Czym niższy czas pomiędzy pobieraniem kolejnych próbek (wyższa częstotliwość samplowania) tym wierniejsze odwzorowanie sygnału.

Odpowiednio przygotowany plik audio stanowi doskonały kontener dla ukrywanych informacji, dzięki zastosowaniom zaawansowanych technik steganograficznych możliwe jest osadzenie dużej ilości danych niejawnych w stosunkowo krótkim pliku audio, przy zachowaniu wysokiego poziomu bezpieczeństwa na wykrycie lub uszkodzenie. Przy zastosowaniu superdokładnych przetworników analogowo - cyfrowych i cyfrowo – analogowych możliwe jest poprawne wyodrębnienie informacji z sygnału analogowego, powstałego z sygnału cyfrowego zawierającego dodatkowe informacje.

Steganografia w dźwięku analogowym jest trudna do zrealizowania, gdyż wymaga realizacji bezpośrednio na sprzęcie o dużej mocy obliczeniowej, będącego w stanie dokonać obliczeń transformaty Fouriera w czasie rzeczywistym. Wyodrębnienie tak osadzonej informacji wymaga porównywalnego sprzętu.

Zastosowanie sygnału cyfrowego, zapisanego na nośnikach w postaci plików jest znacznie prostsze i skuteczniejsze. Transmisja danych za pomocą sygnału

analogowego podlega tym samym ograniczeniom co w przypadku metod steganograficznych opartych na stosie protokołów TCP/IP – niemożliwe jest składowanie sygnału bez uszkodzenia osadzonej w nim zawartości za pomocą nośników analogowych, pominąć tu można urządzenia doskonale czułe, które nie są w użytku powszechnym. Informacja zapisana w cyfrowym pliku audio może być przesyłana na różne sposoby, nie jest konieczne odtworzenie pliku w celu wyodrębnienia informacji – wystarczy fakt jego posiadania.

3.1. HAS (ang. Human Auditory System) - ludzki układ słuchu.

Steganografia w plikach audio stanowi większe wyzwanie niż jej odpowiednik w plikach graficznych. Ludzki system słuchu jest bardziej czuły na zakłócenia spowodowane przez biały szum AWGN (ang. Additive White Gaussian Noise). Zakłócenia te mogą być wykrywalne nawet gdy występują o 70db ciszej od poziomu normalnego dźwięku. Z drugiej strony, układ HAS charakteryzuje się niską czułością na zmiany siły tonu. Oznacza to, że głośne dźwięki przykrywają dźwięki o niższej głośności, przez co stają się one niesłyszalne. Cecha ta jest wykorzystywana min. w kompresji sygnałów dźwiękowych mp3.[15]

MP3 (ang. MPEG Audio Layer-3) jest przykładem kompresji stratnej opracowanej do zmniejszania objętości plików muzycznych. Bazuje na odpowiednio zmodyfikowanej dyskretnej transformacie cosinusowej i modelu psychoakustycznym. Dźwięk skompresowany z przepływnością (ang. bitrate) 128kbps daje zazwyczaj jakość niskiej klasy odtwarzaczom CD. Bitrate rzędu 192kbps jest dla większości ludzi nieodróżnialny od oryginału.[9]

Model psychoakustyczny jest matematycznym modelem, mówiącym jakie informacje o dźwięku są rozpoznawalne dla ludzkiego zmysłu słuchu, przykładowo czułość ludzkiego ucha na dźwięki o różnych częstotliwościach (dźwięki bardzo niskie i bardzo wysokie są niesłyszalne). Zakres częstotliwości słyszalnego dźwięku szacuje się w przedziale od 20hz do 20khz i maksymalną czułość w zakresie 2khz-4khz.

Badając ludzki układ słuchowy zaobserwowano zjawisko maskowania jednych dźwięków przez drugie. Wyróżnia się 3 typy maskowania:

- maskowanie jednoczesne - ciche dźwięki o częstotliwościach zbliżonych do

częstotliwości dźwięku głośnego nie są słyszalne

- maskowanie pobodźcowe - głośny dźwięk potrafi zagłuszyć cichsze dźwięki występujące bezpośrednio po nim
- maskowanie wsteczne - cichy dźwięk, poprzedzający dźwięk głośny nie jest słyszalny.[15]

Pewne niedoskonałości układu HAS są podstawą dla konstruowania systemów steganograficznych. Jest to min. wspomniane wcześniej zjawisko maskowania lub chociażby przyzwyczajenie ludzkie do niskiej jakości dźwięku posiadającego słyszalne zakłócenia. Przyzwyczajenie to może wynikać z niskiej jakości odbiorników radiowych lub telewizyjnych, gdzie występowanie szumów, trzasków lub innych zakłóceń jest rzeczą normalną i już praktycznie nie drażniącą. Umożliwia to dodanie odpowiednio skonstruowanych zanieczyszczeń do sygnału kontenera tak, aby jednocześnie zawierały informację niejawną, ale również nie obniżały znacząco jakości dźwięku.

3.2. Założenia bezpiecznego stegosystemu

Konstruując bezpieczny stegosystem oparty na sygnale audio należy spełnić poniższe warunki:

- każda osadzona informacja powinna być odporna na przetwarzanie pliku kontenera poprzez jego kompresję/dekompresję w stopniu generującym akceptowalną jakość wynikową dźwięku. Niedopuszczalne jest, aby podziałanie na stegosystem kompresją o niewielkim współczynniku redukcji sygnału spowodowało uszkodzenie wiadomości.
- każda wiadomość powinna być odporna na ataki mające na celu jej wykrycie lub uszkodzenie.

W celu uzyskania większego poziomu bezpieczeństwa danych można zastosować zwielokrotnienie ukrywanych danych i każdą z kopii umieścić w innej części pliku kontenera. W przypadku uszkodzenia pliku lub strumienia w trakcie przesyłu danych istnieje wyższe prawdopodobieństwo prawidłowego wyodrębnienia osadzonych informacji, niż w przypadku, gdy ukryta zostanie tylko

jedna kopia wiadomości.

Regulacja wykorzystania pasma w sieciach komputerowych może spowodować, że sygnał strumieniowy audio będzie ulegał uszkodzeniom w trakcie przesyłu w przypadku wysokiego obciążenia łącza. Mechanizm ten jest powszechnie wykorzystywany przy dzieleniu pasma, gdyż u jego założeń leży fakt, że uszkodzenie strumienia multimedialnego jest dopuszczalne i nie wpłynie znacząco na odbiór informacji, czego nie można powiedzieć o zwykłych danych binarnych, które nie mogą ulec uszkodzeniom. Oznacza to, że w przypadku jednoczesnego działania radia internetowego i pobierania np. archiwum, przeznaczane jest szersze pasmo dla pobierania pliku - żaden bajt danych nie może zostać utracony. W przypadku utraty części pakietów z pasma wykorzystanego przez radio, wystąpią jedynie słyszalne zakłócenia, jednak interpretacja sygnału będzie możliwa.

3.3. Wybrane metody

3.3.1. LSB

Zasada działania techniki LSB (ang. Least Significant Bite) w plikach muzycznych jest zbliżona do tej samej metody używanej przy algorytmach bazujących na grafice.

W standardowym algorytmie LSB wybierany jest pewien zbiór danych reprezentujących pojedyncze dźwięki – sampli za pomocą klucza lub określonego z góry algorytmu. Ukrywanie informacji odbywa się poprzez zamianę najmłodszych bitów poszczególnych bajtów wybranego zestawu sampli.

Xj[i]->m[i];

Duży rozmiar plików audio umożliwia umieszczenie w nich dużej ilości danych poprzez manipulowanie najmłodszym bitem. Ze względu na specyficzną cechę układu HAS, manipulowanie najmłodszymi bitami powinno być przeprowadzone ostrożnie, wysoka czułość ludzkiego ucha na biały szum AWGN powoduje, że ilość modyfikowanych bitów nie może być zbyt duża. Powstałe w ten sposób zniekształcenia dźwięku mogą być łatwo wykrywalne. Ogranicza to znacznie ilość możliwych do wykorzystania bitów. Badania [16] wykazały, że wprowadzenie informacji na warstwy LSB powyżej 4 spowoduje znaczną utratę

jakości kontenera i znaczące obniżenie poziomu bezpieczeństwa – zmiany były już dobrze słyszalne.

Wyodrębnienie osadzonych wcześniej danych wykonywane jest poprzez odczyt określonej ilości najmłodszych bitów z zestawu danych opisanych kluczem lub algorytmem.

Pomimo faktu, że możliwe jest osadzenie dużej ilości danych algorytmem LSB, nie oferuje on wystarczającego poziomu bezpieczeństwa danych. Ich uszkodzenie lub całkowite zniszczenie jest możliwe poprzez np. przypadkową zmianę losowo wybranych najmłodszych bitów w całym pliku kontenera – jest to jeden z najprostszych ataków prowadzonych na metodę LSB.

W pracy „Increasing Roustness of LSB Audio Steganography by Reduced Distortion LSB Coding" przeprowadzonej na University of Oulu w Finlandii przez Cvejica i Sappanena zaproponowano zmodyfikowaną wersję algorytmu LSB. Zaproponowany algorytm pozwala na przeniesienie ukrywanych danych z warstwy 4 na warstwę 6. Podstawową różnicą w działaniu jest fakt, że standardowy algorytm zastępuje odpowiedni bit sampla na odpowiedni bit tajnej wiadomości, jeżeli bity te się różnią. Dysponując danymi:
16 bitowa próbka danych o wartości: 0000000000001000
fragment tajnej wiadomości: 00000000
osadzanie danych w 4 warstwie LSB spowoduje, że wartość modyfikowanej próbki danych będzie:
16 bitowa próbka danych po modyfikacji: 0000000000000000.
spowoduje to znaczną zmianę dźwięku, gdyż z wartości 8 spadł on nagle do wartości 0.
Zmodyfikowana metoda LSB umożliwia bezpieczne osadzenie danych nawet na 6 warstwie LSB, poprzez zamianę wartości sampla na najbardziej zbliżony do oryginału.
16 bitowa próbka danych o wartości: 0000000000001000
fragment tajnej wiadomości: 00000000
Wynikiem działania zaproponowanego w [16] algorytmu będzie sygnał:
16 bitowa próbka danych o wartości: 0000000000000111

Jak widać, zmiana dźwięku nastąpiła jedynie o 1 a nie wartość 8 jak w przypadku standardowej techniki. Dzięki temu, można podnieść próg bezpiecznej warstwy dla LSB z 4 na 6, a co za razem idzie podniesienie bezpieczeństwa danych. Mimo przejścia na drugą warstwę kontener nie straci jakościowo tyle, co w przypadku poprzedniej metody. Zamiana sampla na najbardziej zbliżony może zapewnić mniejszą ilość „niezgodności" pomiędzy stegosystemem a oryginałem.

Zaproponowany w [16] algorytm ma postać:

if host sample a>0
if bit 0 is to be embedded
if a i-1 =0 then a i-1 a i-2 ...a 0 =11...1
if a i-1 =1 then a i-1 a i-2 ...a 0 =00...0 and
if a i+1 =0 then a i+1 =1
else if a i+2 =0 then a i+2 =1
...
else if a 15 =0 then a 15 =1
else if bit 1 is to be embedded
if a i-1 =1 then a i-1 a i-2 ...a 0 =00...0
if a i-1 =0 then a i-1 a i-2 ...a 0 =11...1 and
if a i+1 =1 then a i+1 =0
else if a i+2 =1 then a i+2 =0
...
else if a 15 =1 then a 15 =0
if host sample a<0
if bit 0 is to be embedded
if a i-1 =0 then a i-1 a i-2 ...a 0 =11...1
if a i-1 =1 then a i-1 a i-2 ...a 0 =00...0 and
if a i+1 =1 then a i+1 =0
else if a i+2 =1 then a i+2 =0
...
else if a 15 =1 then a 15 =0
else if bit 1 is to be embedded

if a i-1 =1 then a i-1 a i-2 ...a 0 =00...0

if a i-1 =0 then a i-1 a i-2 ...a 0 =11...1 and

if a i+1 =1 then a i+1 =0

else if a i+2 =1 then a i+2 =0

...

else if a 15 =1 then a 15 =0 {618}

Jeżeli algorytm osadzający informację niejawną jest różny w obu odmianach algorytmu LSB, tak mechanizm wyodrębniania jest jednakowy i nie uległ zmianie i polega o odczytaniu bitu z i-tej warstwy LSB.

Podsumowując technikę LSB można stwierdzić że jej pierwsza odmiana wprowadza większą ilość zanieczyszczeń do kontenera, poprzez niekontrolowaną zmianę bitów na określonej warstwie. Wysokość warstwy nie powinna przekraczać czwartej, gdyż spowoduje to znaczne obniżenie jakości kontenera. W przypadku metody zaproponowanej na Univeristy of Oulu wysokość warstwy może zwiększyć się do szóstej, osadzanie danych nie powoduje znacznej modyfikacji sampla, gdyż jego wartość jest obniżana do wartości mu najbliższej przy zmodyfikowanym określonym bicie – czyli próbka dźwięku wynikowa będzie bardzo zbliżona do oryginalnej.

3.3.2. DSSS

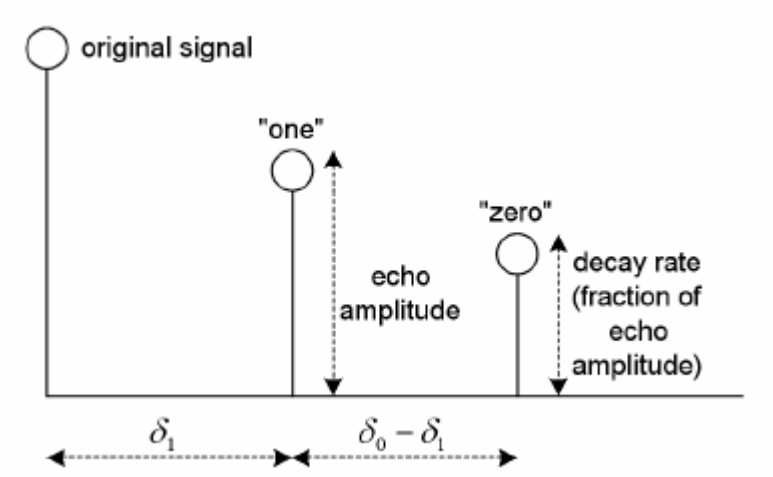
Direct Sequence Spread Spectrum a dokładniej directly carrier-modulated, code sequence modulation) czyli bezpośrednie modulowanie nośnej sekwencją kodową. Jest to jedna z technik rozpraszania widma w systemach szerokopasmowych przy pomocy ciągów kodowych. Jeden ze sposobów działania tej techniki polega na tym, że przy wysyłaniu, strumień danych jest mnożony przez odpowiedni ciąg kodowy o większej szybkości bitowej. W ten sposób wyjściowy strumień informacji zajmuje znacznie szersze pasmo. Dobór ciągu kodowego musi spełniać szereg wymagań. Właściwy jego dobór pozwala na zaszyfrowanie informacji [9]

Technika osadzania danych bazująca na DSSS jest metodą niedoskonałą o

niskim progu odporności danych na uszkodzenia i ataki mające na celu zniszczenie osadzonej informacji. Przyspieszanie lub spowalnianie sygnału audio może skutecznie uszkodzić dane niejawne. Wszelkiego typu konwersje z sygnału analogowego na cyfrowy i odwrotnie mogą zniszczyć osadzone dane. Spowodowane jest to niezgodnością zegarów w konwerterach D/A i A/D.

3.3.3. Dodawanie echa

Wiele technik steganograficznych bazuje na kodowaniu informacji niejawnej poprzez dodawanie echa do sygnału kontenera. Pomimo wysokiej czułości układu HAS na biały szum metoda ta jest skuteczna, gdyż zastosowanie bardzo krótkich odległości pomiędzy sygnałem oryginalnym i echem powoduje traktowanie zakłóceń jako rezonansu. Po dodaniu echa do kontenera okazuje się, że jego właściwości statystyczne pozostają niezmienione.



Rys. 13. Echo hiding

Stegosystem bazujący na dodawaniu echa posiada 4 główne cechy

- amplituda inicjacyjna
- tempo
- przesunięcie „one” (ang. „one” offset)
- przesunięcie „zero” (ang. „zero” offset)[15]

Odległość pomiędzy oryginalnym sygnałem kontenera a dodanym echem jest w pełni zależna od użytego przesunięcia – zero lub jeden.

W procesie osadzania danych sygnał kontenera jest dzielony na mniejsze porcje,

które są traktowane jako niezależne sygnały. Następnie do każdej porcji dodawane jest echo z określonym bitem informacji. Informacja jest osadzana w kontenerze poprzez dodanie echa z przesunięciem 0 lub 1.

Proces wyodrębniania informacji polega na wykryciu odległości pomiędzy sygnałem a echem za pomocą transformaty Fouriera F i odwróconej transformaty Fouriera F^{-1}.

$$F^{-1}\{\log(|F(x)|^2\}.$$

Jeżeli wynik powyższego równania (cepstrum) jest wyższy na pozycji δ_1 niż na pozycji δ_0 wówczas wyodrębniany bit ustawiany jest na wartość 1, w przeciwnym przypadku ustawiany jest na wartość 0.[15]

3.3.4. Modulacja fazy

Algorytmy bazujące na modyfikacji fazy można podzielić na dwie grupy:

- algorytmy wykorzystujące kodowanie fazowe.

  Proste kodowanie fazowe polega na podzieleniu sygnału kontenera na bloki i osadzenie całej informacji niejawnej w spektrum fazowym pierwszego bloku. Jest używane min. przy znakach wodnych (ang. Watermarks)

- algorytmy wykorzystujące modulowanie fazy

  Wykorzystywany jest mechanizm niezależnego wielozakresowego modulowania fazy sygnału. W tym celu używany jest stosunkowo długi blok danych o długości $N = 2^{14}$. Algorytm dokonuje powolnej zmiany fazy w przestrzeni czasowej. Ukrywana informacja, np. znak wodny osadzana jest poprzez dodanie jednej jednostki skali Bark do fazy bloku. Jedna jednostka skali Bark może przechowywać jedynie jeden bit informacji niejawnej.

  Skala Bark jest psychoakustyczną skalą z wartościami z zakresu 0-24 opisującymi 24 słyszalnych zakresów dźwięku. Oblicza się ją wg:

  $$BARK = 13\arctan(0.00076f) + 3{,}5\arctan((\frac{f}{7500})^2)$$

  Gdzie f jest częstotliwością wyrażoną w hercach.

Udowodniono, że sensowny zakres dla osadzanych danych przedstawiony w skali Bark to 0-24 czyli zakres 0-15khz.

Mechanizmy modulowania lub kodowania fazy są technikami niewykorzystującymi

cechy układu HAS, maskowania słabych dźwięków, dźwiękami silniejszymi.[17]

3.3.5. Autokorelacja

Autokorelacja - statystyka opisująca dla danego szeregu czasowego, w jakim stopniu dany wyraz szeregu zależy od wyrazów poprzednich. {wiki}.

Jest to funkcja która dla argumentu k przypisuje wartość współczynnika korelacji Pearsona pomiędzy szeregiem czasowym a tym samym szeregiem cofniętym o k jednostek czasu.

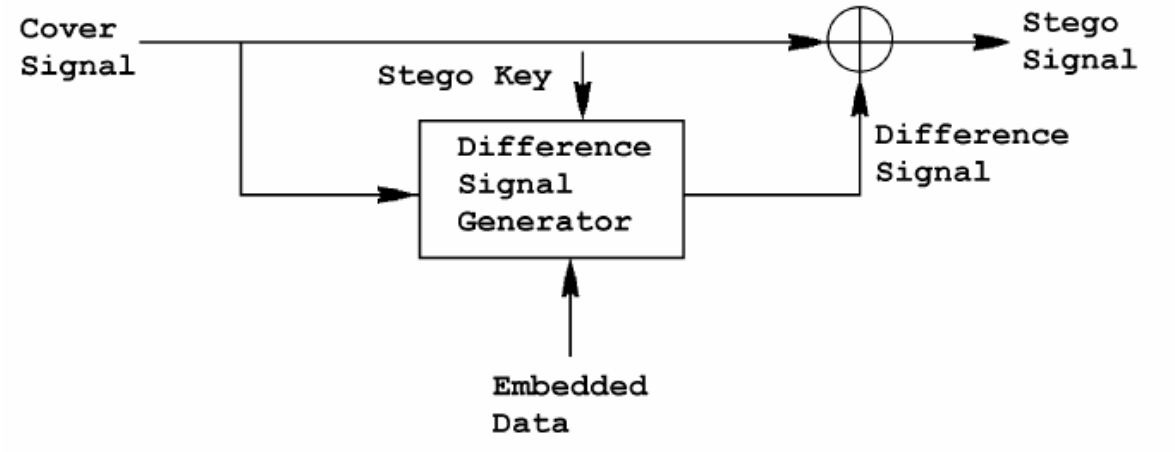
Korelacja Pearsona określa poziom zależności liniowej między zmiennymi losowymi. Niech x i y będą zmiennymi losowymi o ciągłych rozkładach x_i, y_i oznaczają wartości prób losowych zmiennych (i=1,2…n), natomiast $\overline{X}$ i $\overline{Y}$ - wartości średnie tych prób, tj.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{oraz} \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Wówczas współczynnik korelacji liniowej zdefiniowany jest następująco:

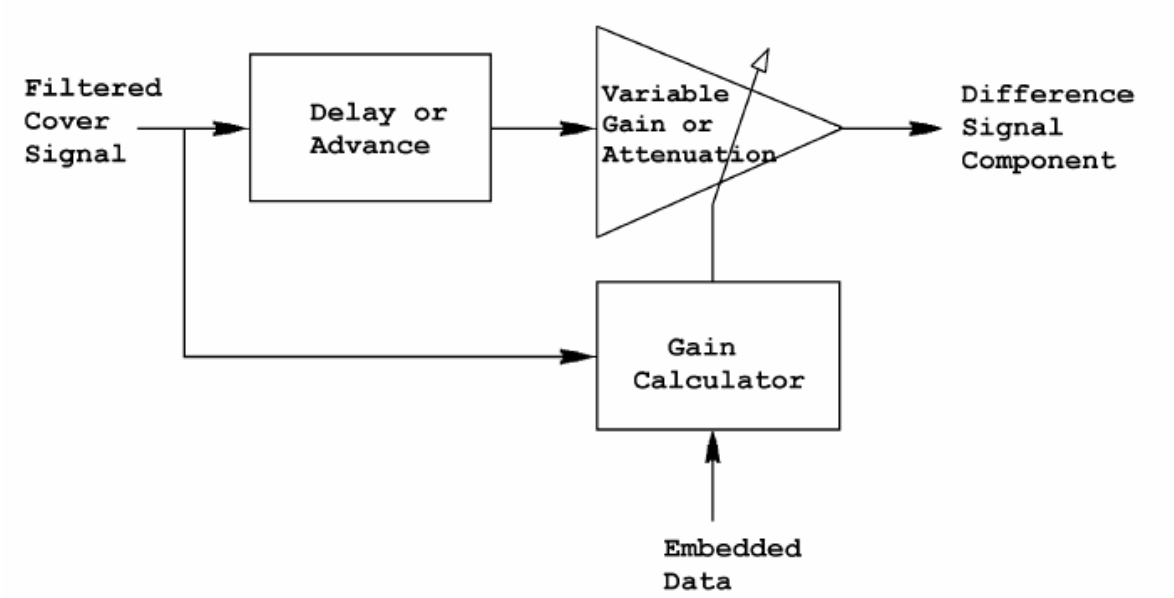
$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Technika manipulowania współczynnikami autokorelacji doskonale nadaje się do osadzania informacji w sygnale analogowym. Należy ona do rodziny modulacji cech statystycznych.[18]



Rys. 14. Diagram blokowy osadzania danych metodą manipulowania współczynnikiem autokorelacji

Wejściowy sygnał kontenera poddawany jest wstępnej filtracji mającej na celu poprawienie własności kontenera. Następnie sygnał ten używany jest do utworzenia specjalnego sygnału zawierającego zmodyfikowane współczynniki autokorelacji. Wykonywane jest to w generatorze sygnału pokazanego na rysunku 13. sygnał ten tworzony jest poprzez dodanie określonej zmiennej opóźnienia. Liczba dodawanych opóźnień jest równa współczynnikowi autokorelacji modulowanego sygnału.



Rys. 15. schemat blokowy generatora sygnału

Krótkoterminowa autokorelacja przefiltrowanego sygnału może przyjmować postać:

$$R(t,\tau) = \int_{t-T}^{t} s(x)s(x-\tau)dx$$

W większości implementacji systemu utrzymuje się, że zmienna g opisująca opóźnienie przyjmuje niewielką wartość, aby opóźnienia były transparentne. Oblicza się je z:

$$g \approx \frac{R_m(t,\tau) - R(t,\tau)}{R(t,2\tau) + R(t-\tau,0)} \text{ lub z } g \approx \frac{R_m(t,\tau) - R(t,\tau)}{R(t,2\tau) + R(t+\tau,0)}$$

Ta technika steganograficzna bazuje na słowach kodowych, które są reprezentowane przez odpowiednio zmodulowane zmiany współczynników autokorelacji, takich jak średnia wartość opóźnień. Istnieje kilka metod reprezentacji słowa kodowego:

1. kodowanie proste, w którym jeden symbol odpowiada dokładnie jednej wartości współczynnika autokorelacji
2. multi-level symbol mapping – każde osadzone słowo kodowe opisane jest skończonym zbiorem wartości współczynników autokorelacji.
3. Manchester symbol encoding – niech sygnał będzie dzielony na dwie równe części, dla każdej z nich zostanie obliczona wartość współczynnika autokorelacji. Następnie różnica pomiędzy tymi dwoma współczynnikami jest modulowana w celu zakodowania danych.
4. delay hoping – poszczególne elementy sygnału zmieniają swoje wartości opóźnienia w zależności od wcześniej predefiniowanej tablicy, której istnienie musi pozostać tajne.[18]

Metoda modulowania współczynników autokorelacji przeznaczona jest przede wszystkim do osadzania niewielkiej porcji danych w transmitowanym sygnale analogowym. Najczęściej wykorzystywana jest do znakowania wodnego sygnału. Oznakowana w ten sposób informacja może być swobodnie przechowywana na nośnikach bez utraty zapisanych niejawnych danych. W przypadku wykorzystania modulacji współczynników autokorelacji w stosunku do sygnału cyfrowego uzyska się stegosystem o wysokim poziomie bezpieczeństwa. Niejawne dane nie są podatne na kompresję, dodanie białego szumu Gaussa AWGN o sile nawet 36db nie spowoduje uszkodzenia danych, zmiana tempa o wartość do 10% nie uszkodzi przekazu.

3.4. Podsumowanie

Dźwięk cyfrowy jest doskonałym medium transmisyjnym dla ukrywania danych. Jego duża pojemność steganograficzna i oferowane bezpieczeństwo danych spowodowały, że pliki dźwiękowe stały się celem zainteresowania steganografów. Niedoskonałości ludzkiego układu słuchowego HAS umożliwiają skuteczne ukrycie danych w dźwięku bez wprowadzania dużych zmian w samym sygnale. Istnieją techniki, których wynikiem działania jest stegosystem o identycznych parametrach

statystycznych jak oryginalny sygnał użyty jako kontener. Istnieją algorytmy pozwalające ukrywać informacje w takich obszarach dźwięku, że ich poziom bezpieczeństwa na wykrycie lub uszkodzenie jest bardzo wysoki. Kompresja stratna nie jest już zagrożeniem dla steganografii. Część algorytmów gwarantuje ochronę osadzonych danych przy użyciu kompresji.

Ilość próbek dźwięku jaka może zostać użyta jako kontener jest bardzo duża, możliwości różnej parametryzacji dźwięku – jego częstotliwości, przepływności czy nawet stylu muzycznego jaki reprezentują umożliwiają bezpieczne ukrycie danych.

4. Analiza metod ukrywania w dźwięku

W tej części pracy zostanie zaprezentowana aplikacja StegoSound, oraz wykonane za jej pomocą analizy. Analizie porównawczej zostały poddane dwa algorytmy osadzania danych w dźwięku, działające na szeregu próbek danych wejściowych.

Celem przeprowadzonych badań jest stwierdzenie która z metod gwarantuje wyższy poziom bezpieczeństwa, oraz sprawdzenie, czy na poziom bezpieczeństwa wpływa styl muzyczny reprezentowany przez poszczególne próbki.

Analizie została poddana amplituda dźwięku.

4.1. Środowisko badań

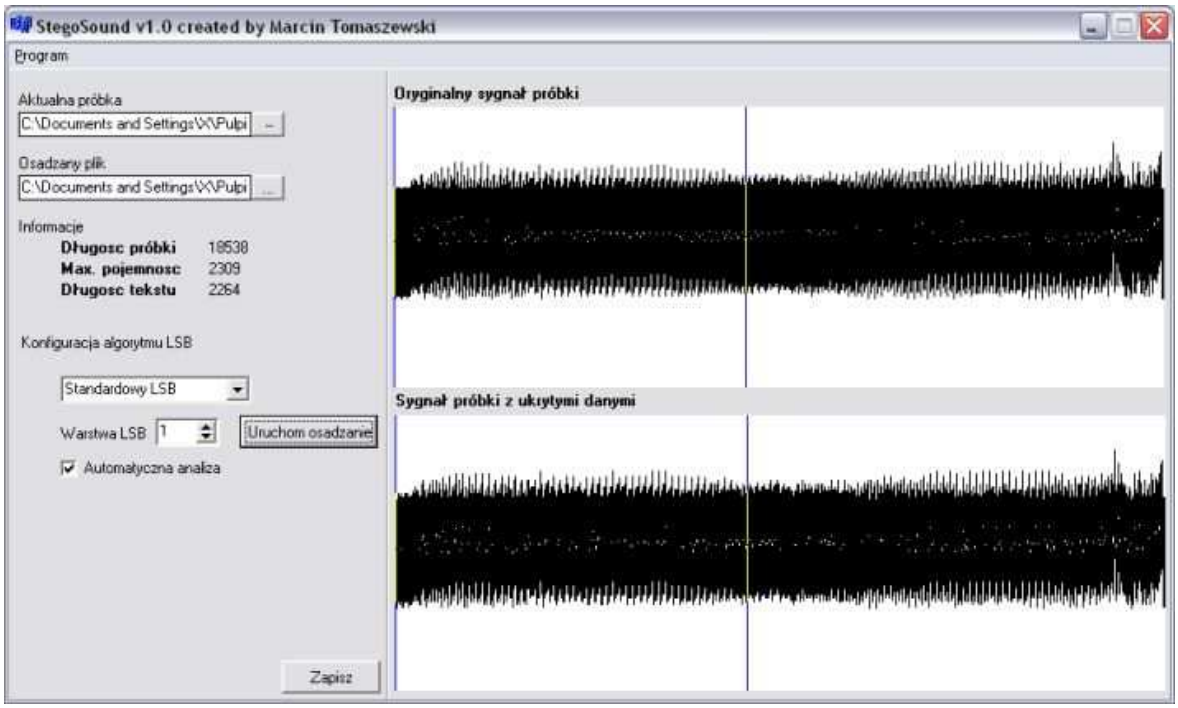
4.1.1. Borland c++ Builder

Środowisko Borland C++ Builder w wersji Personal zostało użyte do napisania aplikacji testującej algorytmy na poszczególnych próbkach danych. Całość projektu została wykonana za pomocą jedynie darmowych i powszechnie dostępnych komponentów do pakietu firmy Borland. Obiekty jakie zostały użyte to: Image, pole edycyjne Edit, Pole tekstowe Label, Pole kombi ComboBox, pole tekstowe CSpinEdit, checkbox, pasek postępu CGauge oraz przyciski BitBtn.

Projekt nie wykorzystuje żadnych zewnętrznych bibliotek, i dzięki kompilacji wraz z wymaganymi modułami jest możliwy do uruchomienia na komputerach na których nie zainstalowano pakietu firmy Borland.

4.1.2. Aplikacja StegoSound

Aplikacja StegoSound jest narzędziem umożliwiającym przeprowadzenie analiz porównawczych dwóch badanych algorytmów steganograficznych bazujących na sygnale dźwiękowym.



Rys. 16. Aplikacja testowa StegoSound

Aplikacja umożliwia wczytanie pliku kontenera oraz pliku zawierającego tekst jawny przeznaczony do osadzenia. Na podstawie wyboru metody oraz warstwy bitowej na jakiej ma działać program dokonuje wpisania tekstu do sygnału dźwiękowego a następnie generuje wykresy obu sygnałów:

- oryginalnego przed osadzeniem danych tekstowych
- zmodyfikowanego, zawierającego ukryte dane

Do operacji na poszczególnych bitach została użyta prosta klasa napisana w C++ która ma postać:

```
class bin
{
public:
        char *CharToBin(int);
        void SetChar(char*);
        char BinToChar();
        char *SetBit(int);
      char *SetBit2(int,int);
        char GetBit(int);
private:
        char *chars;   //wartosc bitowa znaku
};


//metoda zapisujaca wartosc binarna do obiektu
//do dalszej obrobki
void bin::SetChar(char *_liczba)
```

```
{
chars=new char[strlen(_liczba)-1];
strcpy(chars,_liczba);
}

//metoda zamieniajacy znak na zapis binarny
char *bin::CharToBin(int liczba)
{
chars=new char[9];
int tmp=0;
for(int i=7;i>-1;i--)
        {
        tmp=liczba%2;
        if(tmp==1) chars[i]='1';
        else     chars[i]='0';
        tmp=0;
        liczba=liczba/2;
      chars[8]='\0';
        }
return(chars);
}


//metoda zamieniajaca zapis binarny na znak
char bin::BinToChar()
{
int tmp=0;
int mnoznik=128;
int liczba=0;
for(int i=0;i<8;i++)
        {
        liczba=(chars[i]-48)*mnoznik;
        tmp=tmp+liczba;
        mnoznik=mnoznik/2;
        }
return tmp;
}

//metoda ustawiajaca najmlodszy bit na bit przeciwny
char *bin::SetBit(int n)
{
if(chars[n]=='1')
        chars[n]='0';
else
        chars[n]='1';
return chars;
}

char *bin::SetBit2(int n, int wart)
{
if((chars[n]=='1')&&(wart==1));
else if((chars[n]=='1')&&(wart==0))
```

```
        chars[n]='0';
else if((chars[n]=='0')&&(wart==1))
        chars[n]='1';
else if((chars[n]=='0')&&(wart==0));
}

//metoda zwracajaca najmlodszy bit
char bin::GetBit(int n)
{return chars[n];}
```

Wykorzystanie metod zawartych w klasie bin możliwe było skuteczne manipulowanie poszczególnymi bitami zarówno dźwięku jak i tekstu, za pomocą których dane zostały osadzone do kontenera.

Aplikacja po wczytaniu danych wejściowych dokonuje analizy pojemności kontenera – jeżeli informacji przeznaczonych do ukrycia jest więcej niż maksymalna pojemność kontenera, program uniemożliwi dalsze operacje.

4.2. Badane algorytmy

Stworzony program StegoSound dokonuje analizy dwóch odmian algorytmu Least Significant Bit – LSB:

- standardowa metoda LSB polegająca na zamianie wartości bitu kontenera na określonej warstwie w zależności od wartości bitu ukrywanych danych
- zmodyfikowana metoda LSB polegająca dodatkowo na zamianie bitów na niższych warstwach w celu możliwie jak najmniejszego zmodyfikowania wartości próbki dźwięku. Metoda opracowana na Univeristy of Oulu.

W dalszej części pracy będą określone odpowiednio jako:

- Algorytm 1
- Algorytm 2

4.3. Próbki danych

Analizie zostało poddanych 5 próbek dźwięku o małej długości (poniżej sekundy) reprezentujące różne style muzyczne.

numer próbki	Długość	częstotliwość próbkowania	głębia bitów	Kanały	Autor	Styl muzyczny	Utwór
1	0,417	44,1	8	mono	Saturnus	Doom Metal	I Long
2	0,509	44,1	8	mono	Seven Main Sins	Death Metal	Kamienna Tablica
3	0,601	44,1	8	mono	Omega	Rock	Dziewczyna o perłowych włosach
4	0,546	44,1	8	mono	Puff Dady	Rap	Godzilla
5	0,493	44,1	8	mono	Eugenia Vlasova & Andru Donalds	Nowoczesna	Wind of hope

Każda próbka dźwięku została pobrana z pierwszej sekundy każdego utworu.
Jako danych podlegających osadzaniu użyto jednej strony niniejszej pracy.
Przeprowadzone analizy przebiegały:

- Każda próbka dźwięku została poddana działaniu dwóch algorytmów
- Każdy algorytm został przetestowany na dwóch warstwach bitowych – pierwszej i czwartej
- Do każdej próbki danych został wpisany ten sam tekst

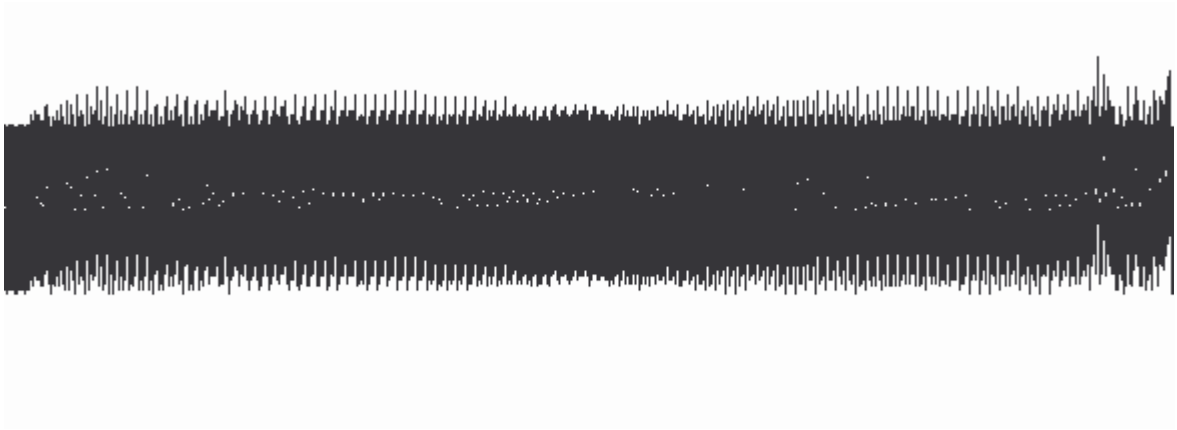
4.4. Wyniki poszczególnych analiz

Wyniki analiz prezentowane są wg schematu:

- Numer badania
- Numer próbki
- Algorytm
- Warstwa bitowa
- Wykresy
- komentarz

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
1	1	1	1

Amplituda dźwięku przed wpisaniem danych

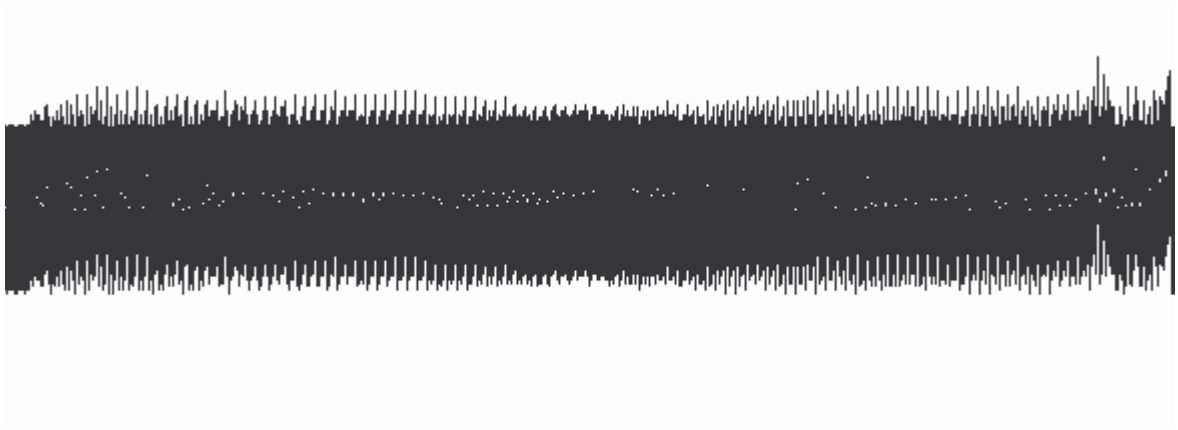


Amplituda dźwięku z wpisanymi danymi

Komentarz:

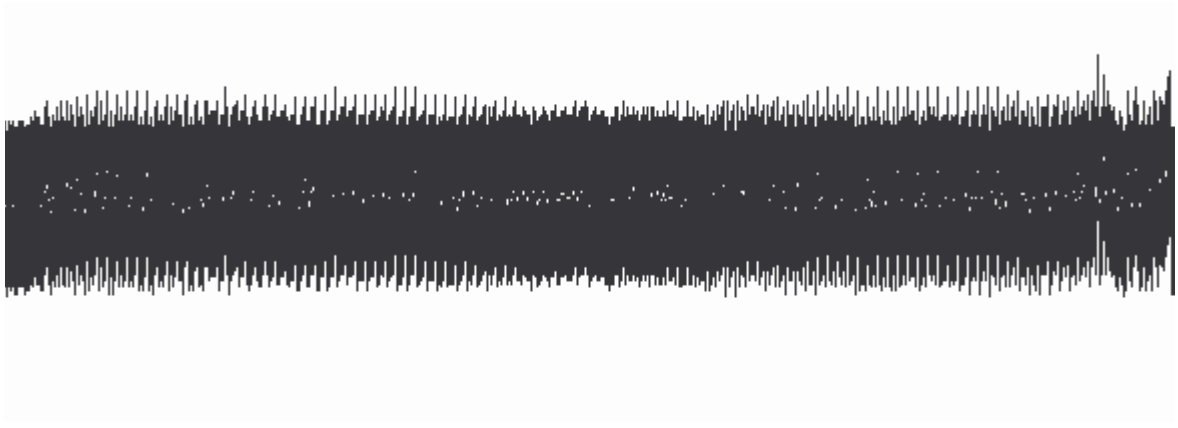
Widoczne niewielkie odkształcenia sygnału w pierwszej części wykresu amplitudy. Zniekształcenia nieznaczne.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
2	1	2	1



Amplituda dźwięku przed wpisaniem danych

Amplituda dźwięku z wpisanymi danymi

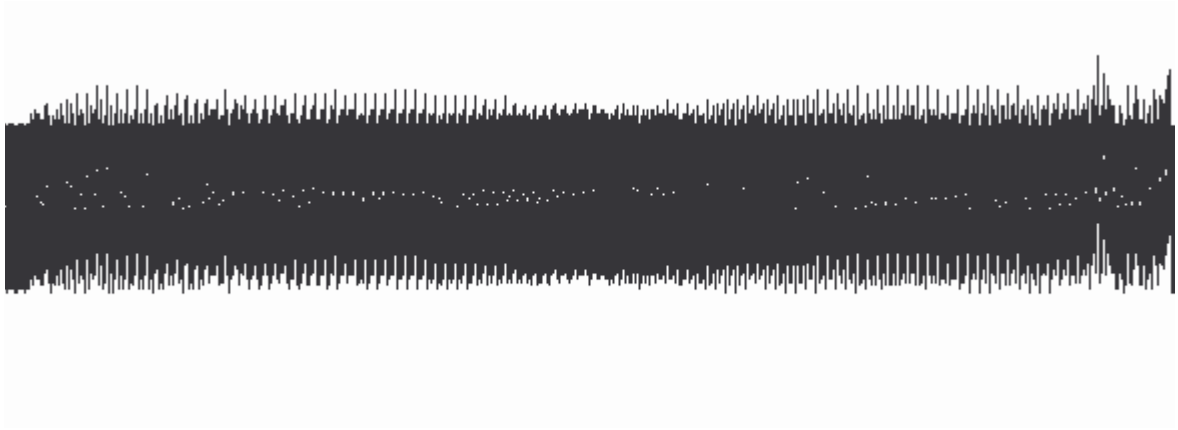


Komentarz:

Niewielkie zniekształcenia widoczne w początkowej części sygnału. Dźwięk wynikowy identyczny jak w przypadku algorytmu 1 działającego na tej samej warstwie.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
3	1	1	3

Amplituda dźwięku przed wpisaniem danych



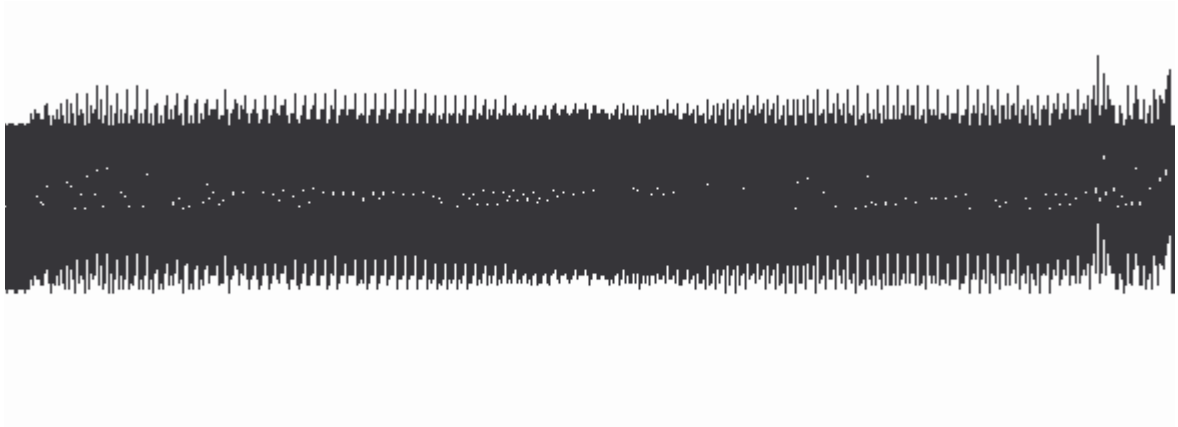
Amplituda dźwięku z wpisanymi danymi

Komentarz:

Widoczne mocne zniekształcenia sygnału na całej długości. Bardzo duże różnice między oryginałem a sygnałem wynikowym.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
4	1	2	3

Amplituda dźwięku przed wpisaniem danych



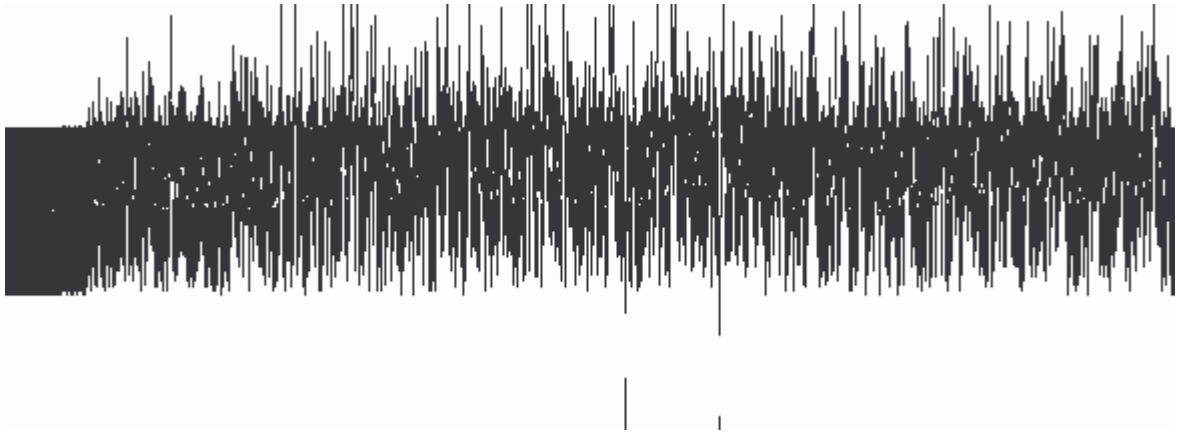
Amplituda dźwięku po wpisaniu danych

Komentarz:

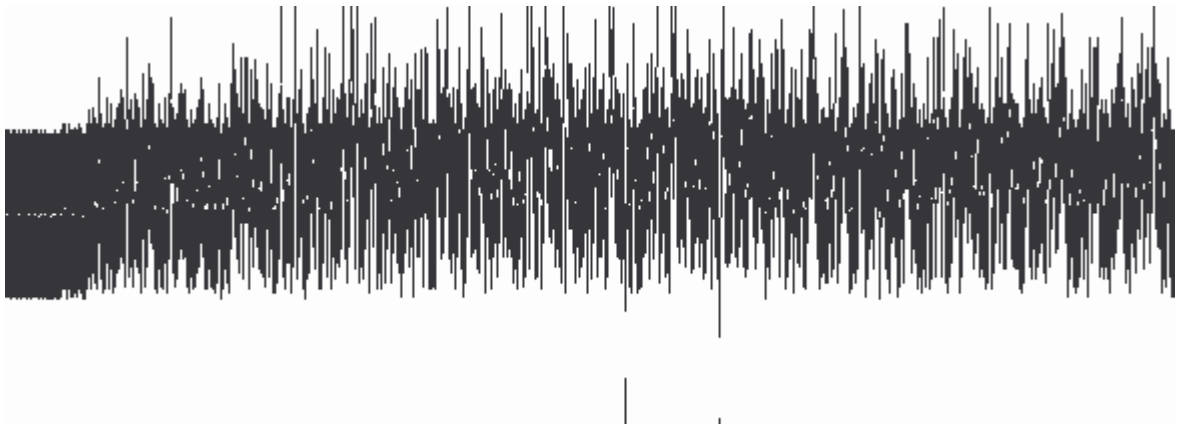
Duże zniekształcenie sygnału dźwiękowego. W stosunku do algorytmu 1 zmiany dokonane w procesie ukrywania danych są znacznie mniejsze. Zwłaszcza jest to widoczne w środkowej części sygnału.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
5	2	1	1

Amplituda dźwięku przed wpisaniem danych



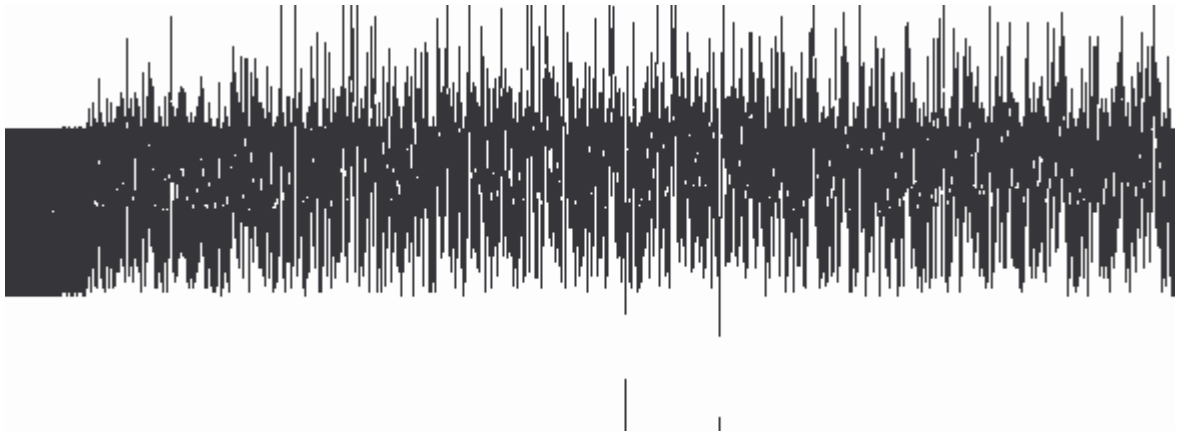
Amplituda dźwięku z wpisanymi danymi



Komentarz:

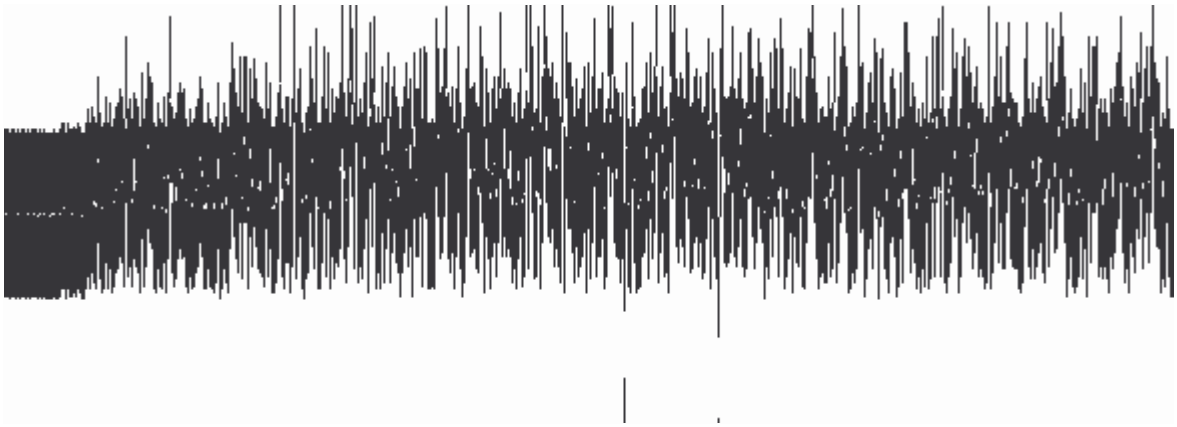
Zmiany widoczne praktycznie tylko w początkowej części sygnału. Reszta sygnału z powodu dużych skoków amplitudy jest utrudniona w analizie.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
6	2	2	1



Amplituda dźwięku przed wpisaniem danych

Amplituda dźwięku z wpisanymi danymi

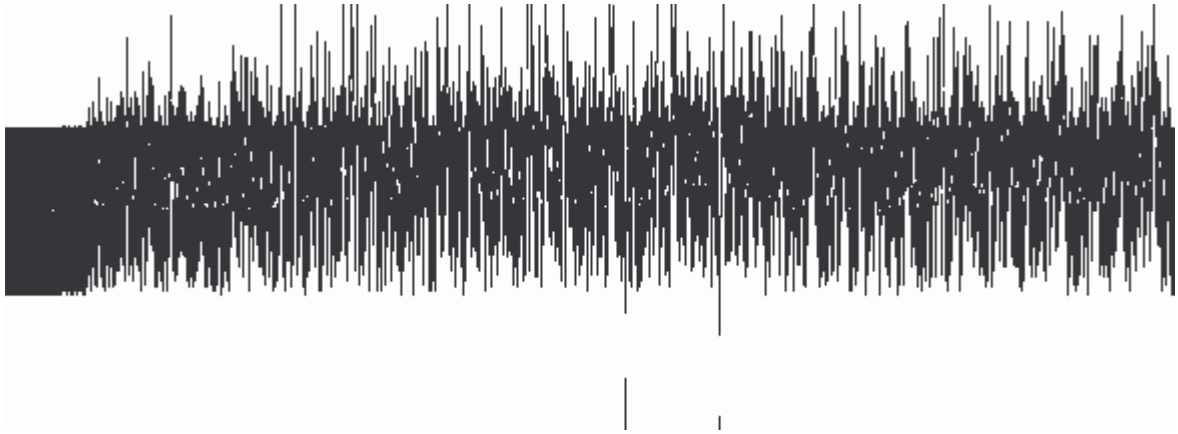


Komentarz:

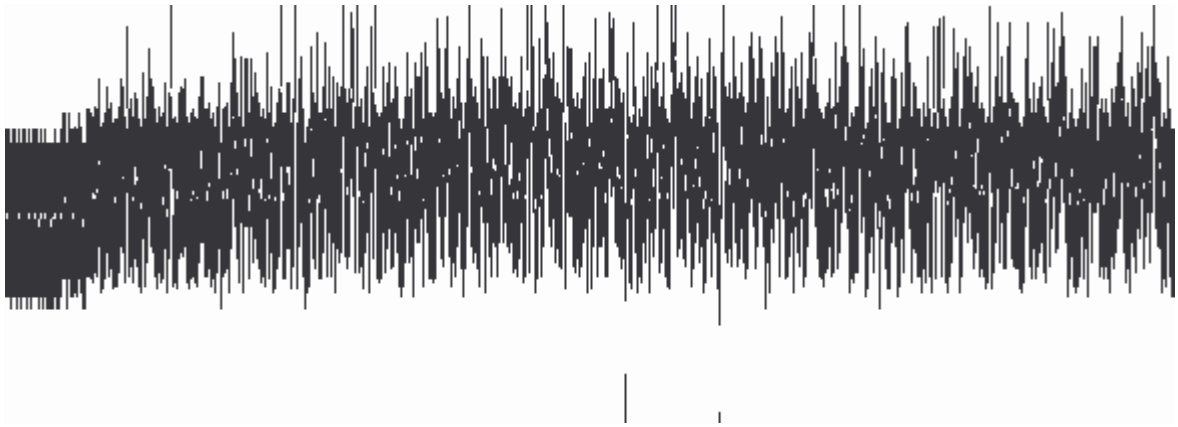
Zmiany widoczne jedynie w początkowej części sygnału. Zmiany dokonane przez algorytm identyczne jak w przypadku algorytmu 1.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
7	2	1	3

Amplituda dźwięku przed wpisaniem danych



Amplituda dźwięku z wpisanymi danymi

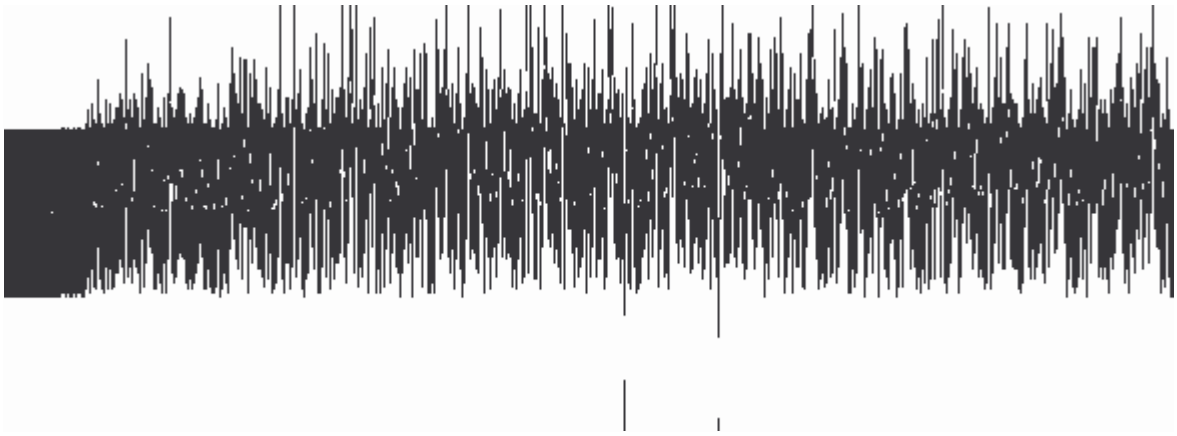


Komentarz:

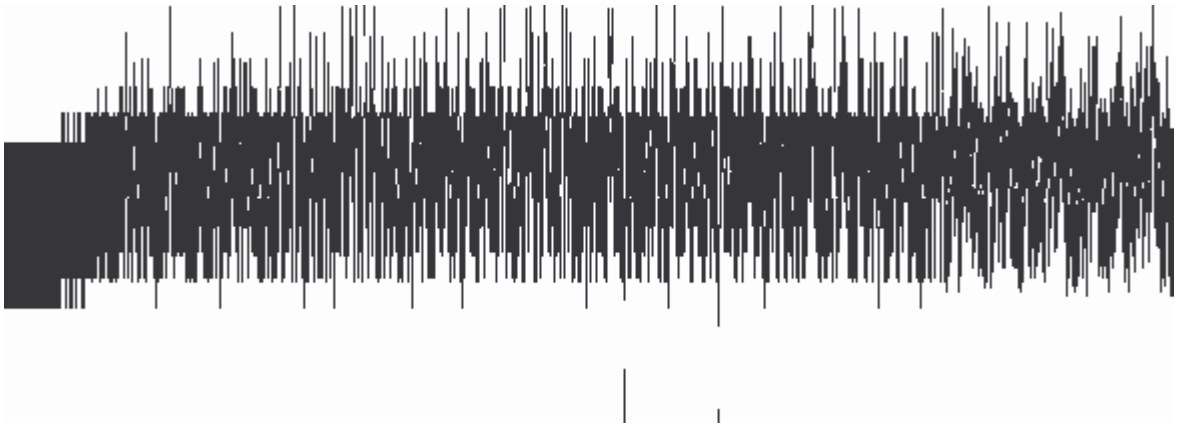
Zmiany sygnału widoczne w początkowej części dźwięku. Pozostała część również zmodyfikowana, jednak analiza utrudniona przez częstą zmianę amplitudy.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
8	2	2	3

Amplituda dźwięku przed wpisaniem danych



Amplituda dźwięku z wpisanymi danymi



Komentarz:

Niewielkie zmiany sygnału na całej długości.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
9	3	1	1

Amplituda dźwięku przed wpisaniem danych

Amplituda dźwięku z wpisanymi danymi

Komentarz:

Zmiany sygnału widoczne na całej długości – spowodowane niewielką zmianą amplitudy.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
10	3	2	1

Amplituda dźwięku przed wpisaniem danych

Amplituda dźwięku z wpisanymi danymi

Komentarz:

Modyfikacja identyczna jak w przypadku algorytmu 1.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
11	3	1	3

Amplituda dźwięku przed wpisaniem danych

Amplituda dźwięku z wpisanymi danymi

Komentarz:

Widoczne mocna modyfikacja sygnału. Ostatnia część dźwięku niezmodyfikowana – spowodowane dużą różnicą między długością osadzanych danych a maksymalną pojemnością kontenera.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
12	3	2	3

Amplituda dźwięku przed wpisaniem danych

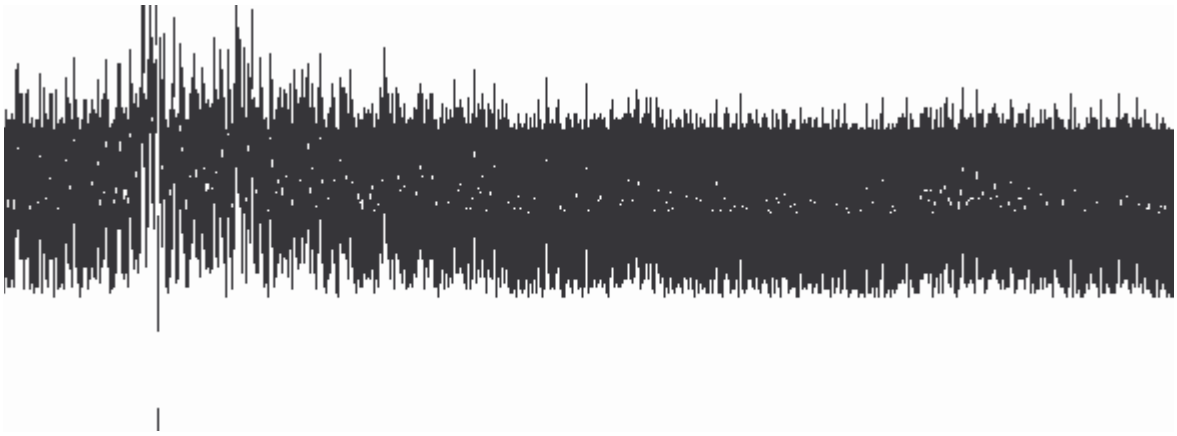
Amplituda dźwięku z wpisanymi danymi

Komentarz:

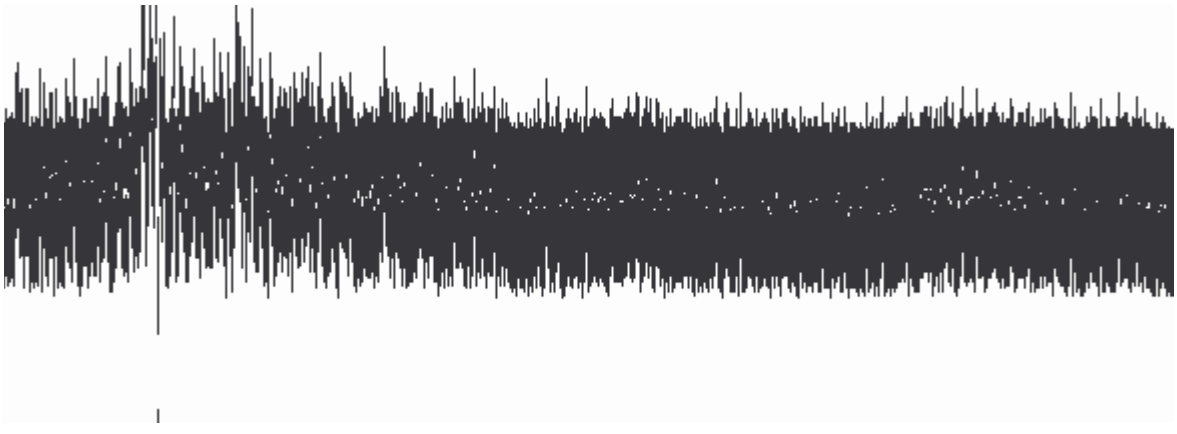
Bardzo mocno zarysowane różnice w sygnale

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
13	4	1	1

Amplituda dźwięku przed wpisaniem danych



Amplituda dźwięku z wpisanymi danymi

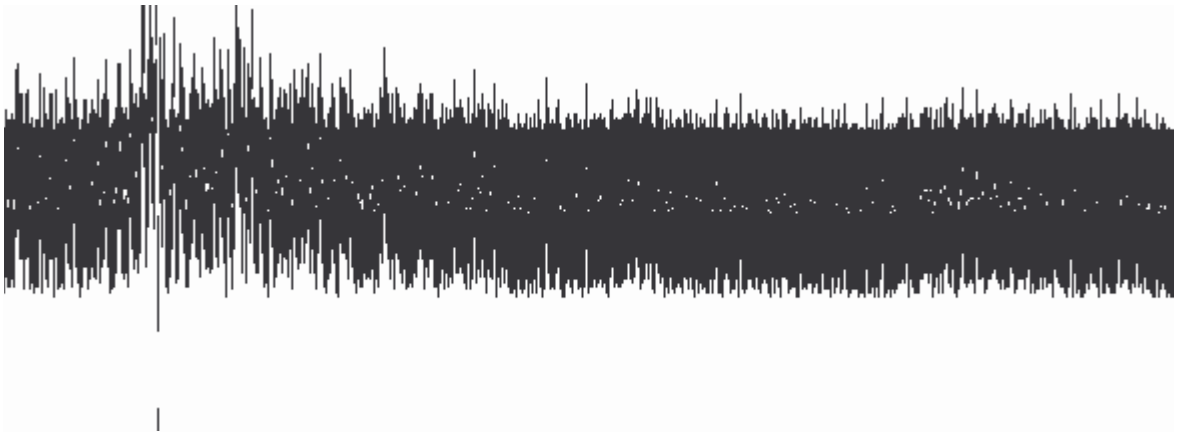


Komentarz:

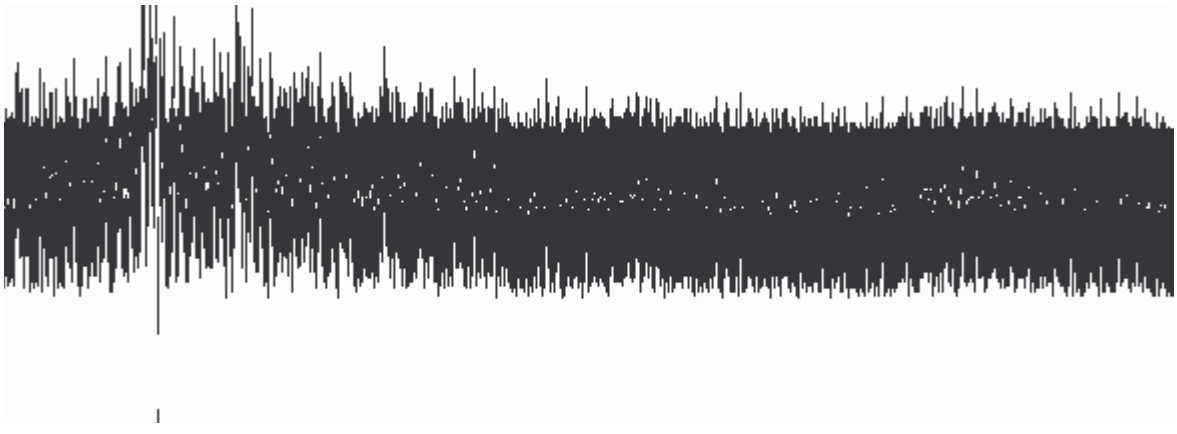
Zmiany praktycznie niewidoczne

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
14	4	2	1

Amplituda dźwięku przed wpisaniem danych



Amplituda dźwięku z wpisanymi danymi

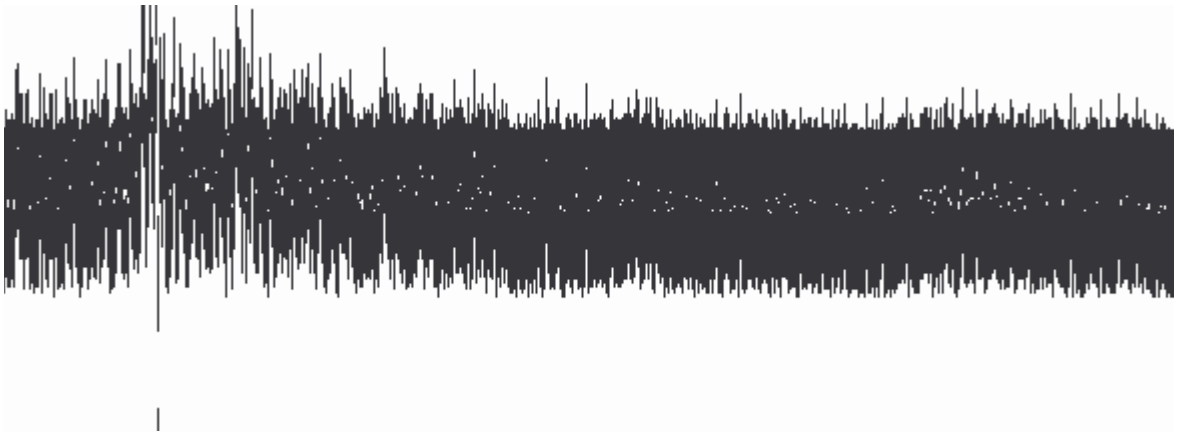


Komentarz:

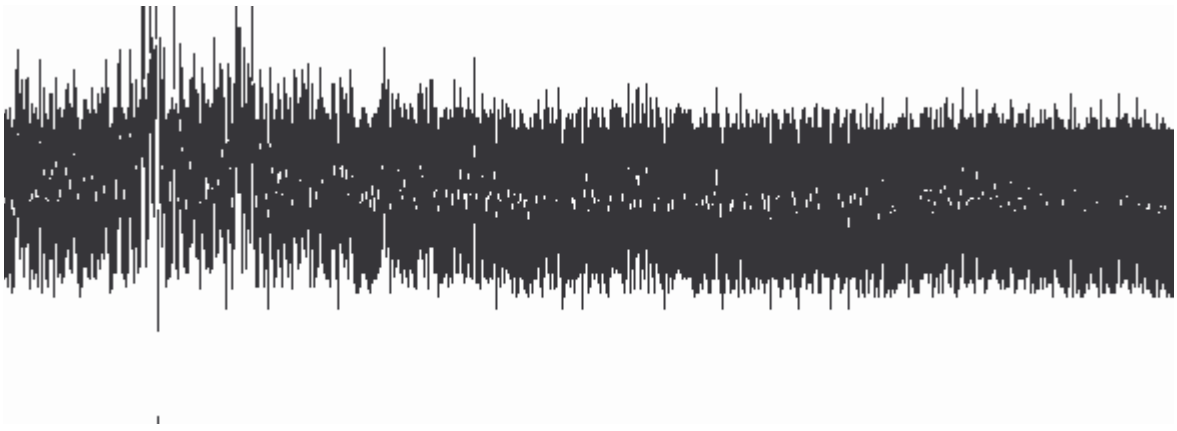
Zmiany praktycznie niewidoczne – zupełnie jak w przypadku algorytmu 1.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
15	4	1	3

Amplituda przed wpisaniem danych



Amplituda dźwięku z wpisanymi danymi

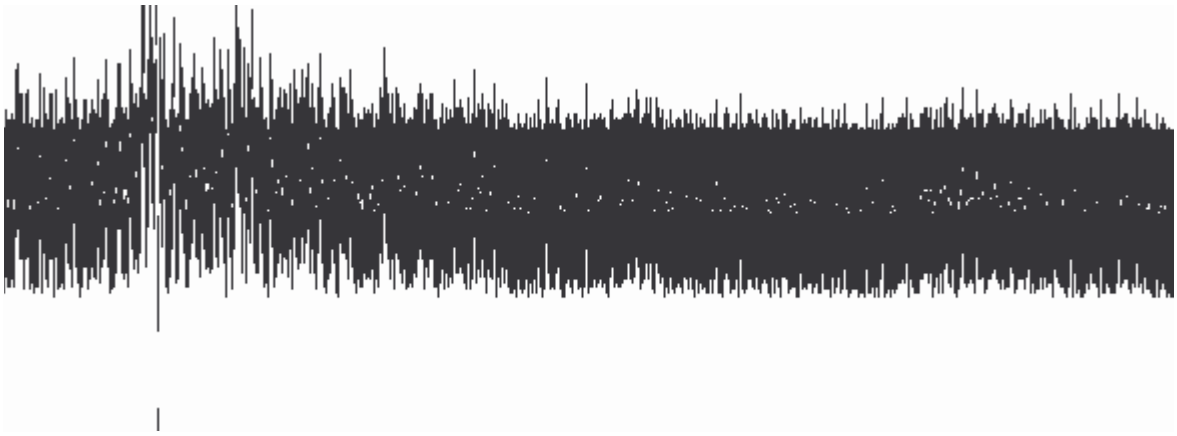


Komentarz:

Zmiany mało widoczne, jednak w stosunku do warstwy 1 modyfikacje są większe.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
16	4	2	3

Amplituda dźwięku przed wpisaniem danych



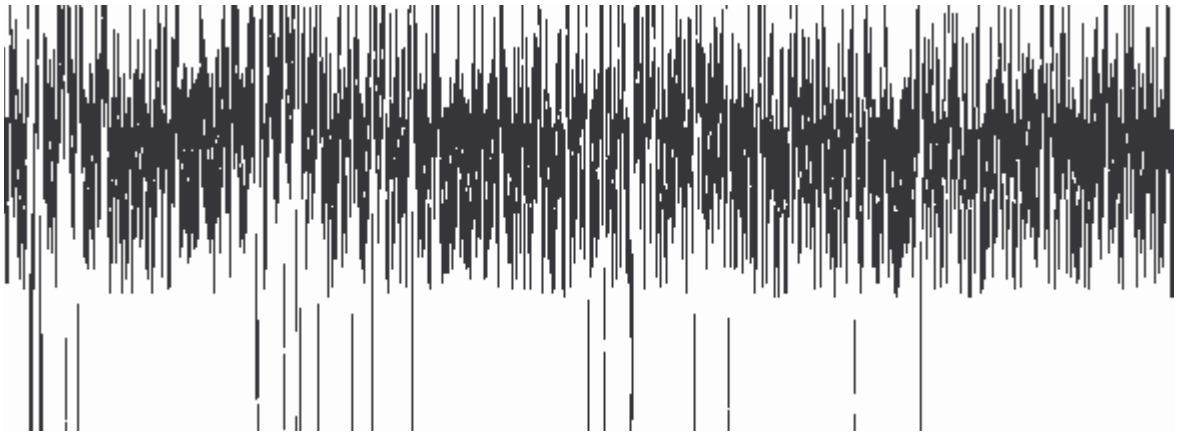
Amplituda dźwięku z wpisanymi danymi

Komentarz:

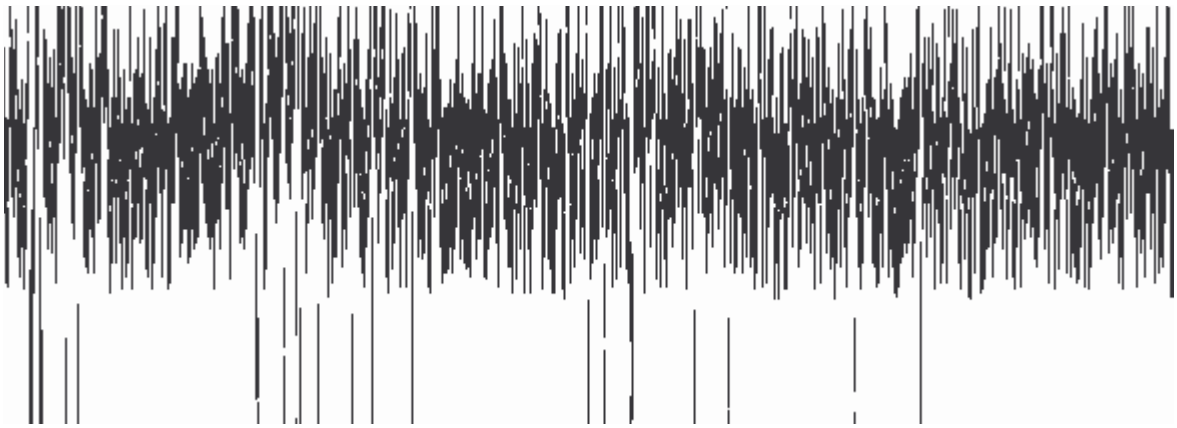
Modyfikacje mało widoczne – znacznie mniej zmian niż w przypadku algorytmu 1.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
17	5	1	1

Amplituda dźwięku przed wpisaniem danych



Amplituda dźwięku z wpisanymi danymi

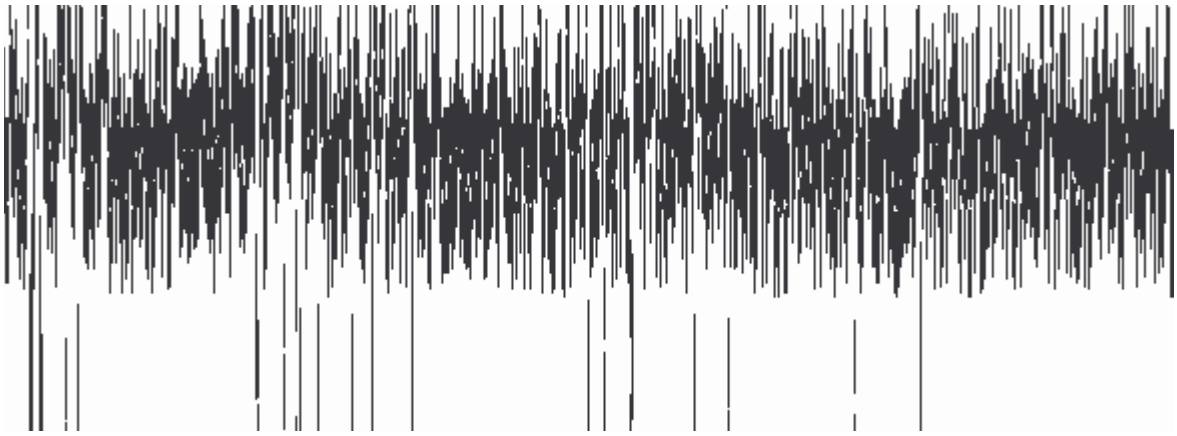


Komentarz:

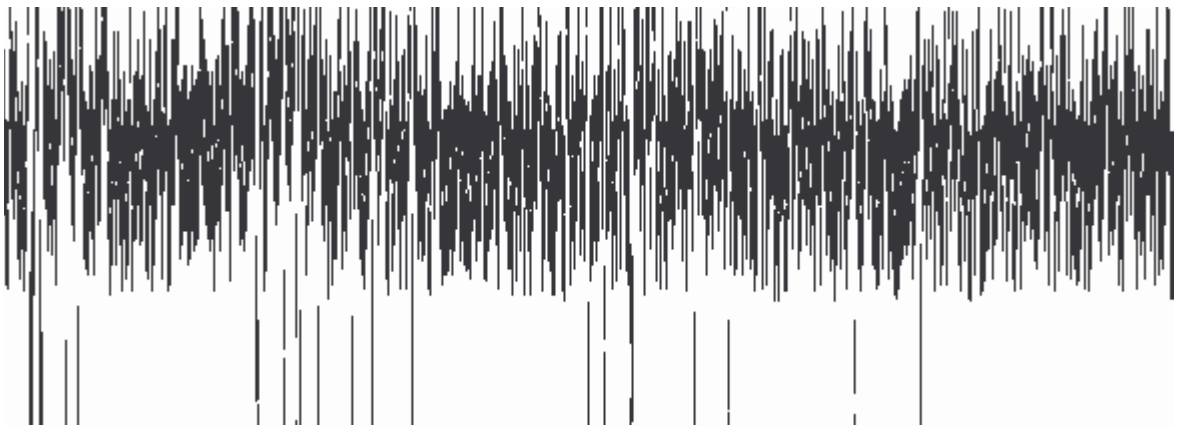
Ze względu na bardzo duże skoki amplitudy modyfikacje sygnału niewidoczne zupełnie.

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
18	5	2	1

Amplituda dźwięku przed wpisaniem danych



Amplituda dźwięku z wpisanymi danymi

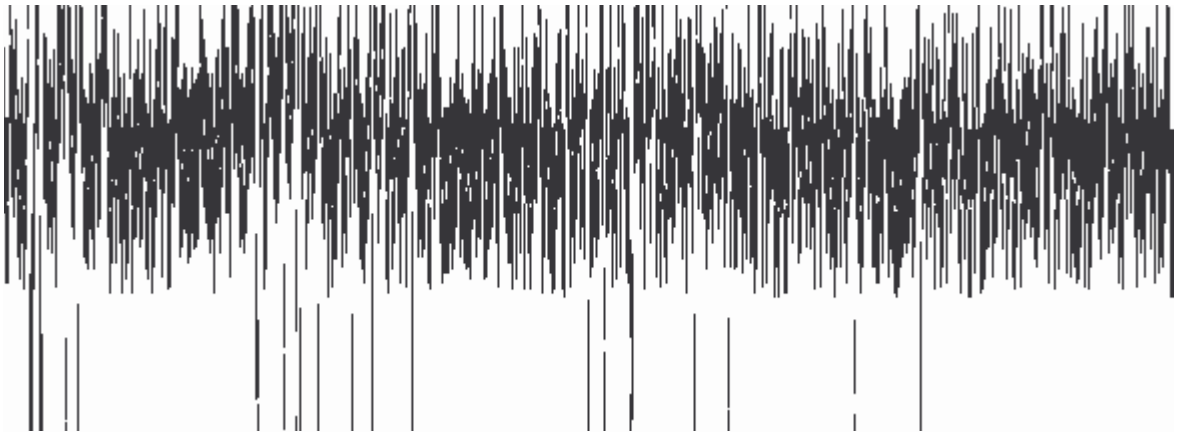


Komentarz:

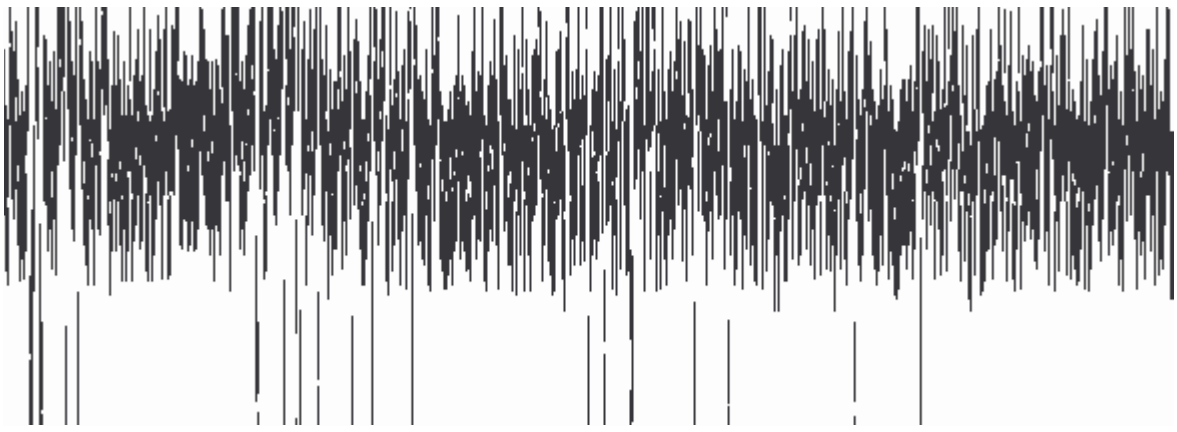
Ze względu na bardzo duże skoki amplitudy zmiany niewidoczne

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
19	5	1	3

Amplituda dźwięku przed wpisaniem danych



Amplituda dźwięku z wpisanymi danymi

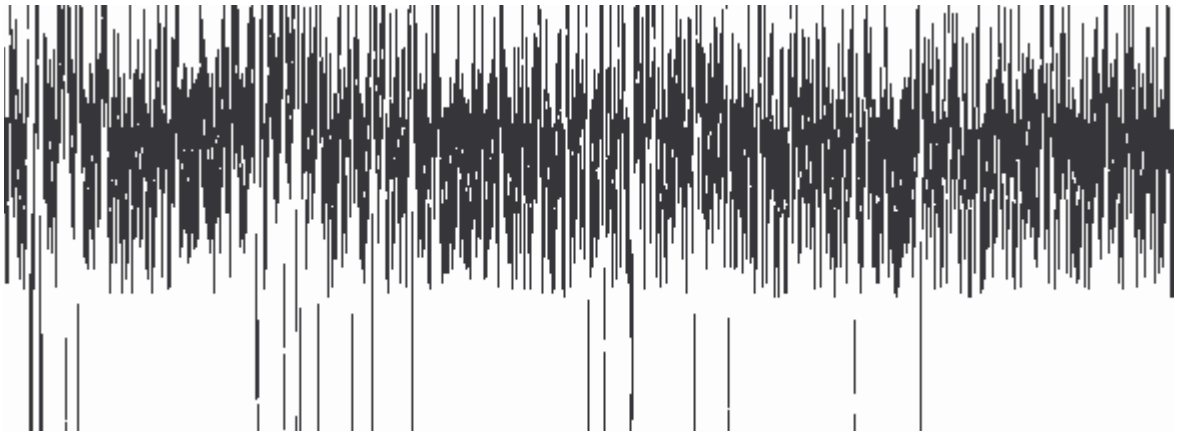


Komentarz:

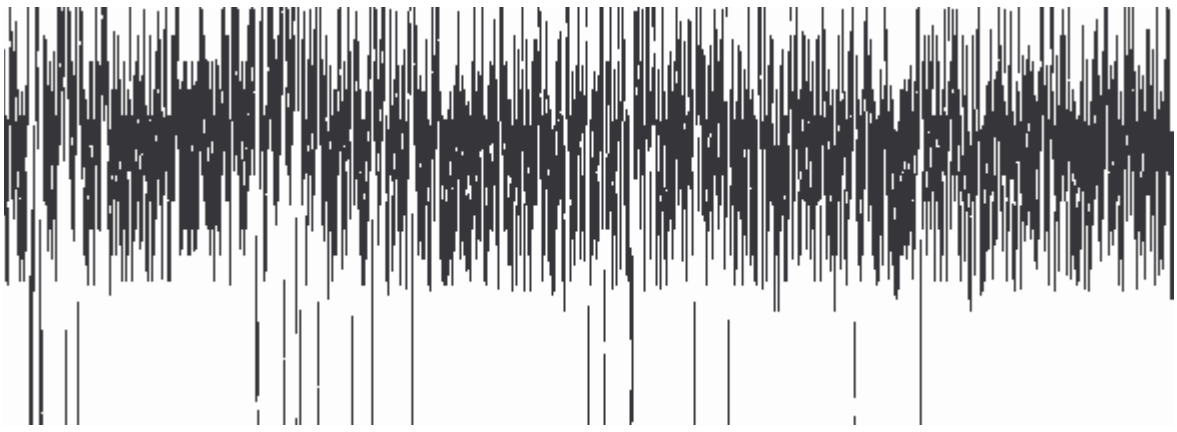
Zmiany minimalne, ale jednak widoczne

Badanie nr	Numer próbki	Algorytm	Warstwa bitowa
20	5	2	3

Amplituda dźwięku przed wpisaniem danych



Amplituda dźwięku z wpisanymi danymi



Komentarz:

Zmiany mniej widoczne w porównaniu do algorytmu 1

Zakończenie i wnioski

Celem pracy było zaprezentowanie istniejących technik steganograficznych, oraz dokonanie analizy porównawczej dwóch wybranych technik ukrywania danych w sygnale dźwiękowym.

Z przeprowadzonych badań wynika, że modyfikacja algorytmu LSB opracowana na University of Oulu generuje mniejsze zakłócenia sygnału, przez co zapisana w kontenerze informacja jest bezpieczniejsza. Algorytmy porównywane były za pomocą tych samych danych wejściowych i identycznych ustawieniach konfiguracyjnych programu testowego StegoSound.

Kolejnym wnioskiem, jaki można wysnuć analizując wyniki przeprowadzonych 20 badań jest taki, że niezależnie od użytego algorytmu ani warstwy bitowej informacja jest bezpieczniejsza w dźwięku o dużych skokach amplitudy niż w dźwięku o małej zmianie. Można więc przypuszczać, że osadzając dane w muzyce, najdoskonalszym medium będą nagrania stylów muzycznych takich jak metal, jazz, czy muzyka elektroniczna. Ballady lub delikatna muzyka klasyczna może nie sprostać wymaganiom stawianym bezpiecznemu kontenerowi. Różnica w amplitudzie łagodnej muzyki jest o tyle mała, że wszelkiego typu zmiany mogą być łatwiej zauważone.

Wprowadzanie zakłóceń do nagrania wymienionego jako gatunek bezpieczny gwarantuje, że akustycznie bardzo trudno będzie rozróżnić oryginał od stegosystemu, lub nawet nie zauważyć występujących zakłóceń. Ostatni przypadek może zdarzyć się przy muzyce metalowej.

Bibliografia

[1] Duncan Sellars, An Introduction To Steganography, http://www.cs.uct.ac.za/courses/CS400W/NIS/papers99/dsellars/stego.html

[2] Nicholas J. Copper, John Langford, Luis von Anh, Provably secure steganography, 2002

[3] Cristian Cachin, AnInformation-Theoretic Model for Steganography, MIT Labs, 1998

[4] Stefan Katzenbeisser, Fabien A. P. Petitcolas, Information Hiding – techiques for steganography and digital watermarking, wydawnictwo Artech House, Boston 1999

[5] Reinhard Wobst, Kryptologia. Budowa I łamanie zabezpieczeń, wydawnictwo RM, Warszawa 2002

[6] J. T. Brassil, Electronic Marking and Identification Techniques to Discourage Document Copying, IEEE Journal on selected areas in communicatons, vol 13, no. 8, October 1995

[7] J. T. Brassil, Document Marking and Identification using Both Line and Word Sifting, AT&T Bell Laboratories, Murray Hill NJ 07974

[8] Luther Deyo, Steganography

[9] http://pl.wikipedia.org

[10] Niels Provos, Peter Honeyman, Hide and Seek – Introduction to Steganography, University of Michigan, IEEE Computer Society, 2003

[11] W. Bender, D. Gruhl, N. Morimoto, and A. Lu., Techniques for data hiding, IBM Systems Journal, vol 35, NOS 384, 1996

[12] M. Celik, G. Sharma, M. Teklap, Universal Image Steganalysis using rate-distortion curves, University Istanbul 2004

[13] autor anonimowy, Internet Agresja i Ochrona, wydawnictwo Robomatic, Wrocław 1998

[14] Kamran Ahsan, Covert Channel Analysis and Data Hiding in TCP/IP

[15] N.Cvejic, Algorhitms for audio watermaring and steganography, Oulu 2004

[16] N. Cvejic, T. Sappanen, Incerasing robustness of lsb audio by reduced distortion LSB Coding, Oulu University

[17] C. Matthews, Behind the music: principles of audio steganography, 2003

[18] R. Petrovic, J. Winograd, K. Jemili, E. Metois, Data Hiding Whitin Audio Signal, 1999 Yugoslavia