STEGANALİZ YAKLAŞIMLARININ KARŞILAŞTIRILMASI

Muhammad D.HASSAN

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

EKİM 2008
ANKARA

Muhammad D.HASSAN tarafından hazırlanan STEGANALİZ YAKLAŞIMLARININ KARŞILAŞTIRILMASI adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Prof.Dr. Şeref SAĞIROĞLU …….
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof.Dr.İrfan KARAGÖZ …….
Elektrik-Elektronik Mühendisliği Anabilim Dalı, G.Ü

Prof.Dr. Şeref SAĞIROĞLU …….
Bilgisayar Mühendisliği Anabilim Dalı, G.Ü

Doç.Dr.M.Ali AKCAYOL …….
Bilgisayar Mühendisliği Anabilim Dalı, G.Ü

Tarih: 23/10/2008

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nail ÜNSAL …….
Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Muhammad D.HASSAN

STEGANALİZ YAKLAŞIMLARININ KARŞILAŞTIRILMASI
(Yüksek Lisans Tezi)

Muhammad D.HASSAN

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Ekim 2008

ÖZET

Günümüzde teknolojinin gelişmesi beraberinde dijital ortamlarda bulunan verilerin güvenliğinin sağlamasını da beraberinde getirmektedir. Şifreleme ve steganografi teknikleri bu güvenliğin sağlanmasında kullanılan çözüm tekniklerini içermektedir. Bu tekniklerin sağlamlığı da analiz teknikleriyle test edilmektedir. Bu çalışmada, resim dosyaları ve ses dosyaları içerisinde saklanan gizli bilgiyi ortaya çıkarmak için kullanılan steganaliz metotları incelenmiş ve bu yöntemlerden elde edilen sonuçlar karşılaştırılmıştır. Ayrıca, bu çalışmada steganaliz teknikleri, yazılımları ve yaklaşımları ile Ki-kare tekniği incelenmiş ve detaylı olarak açıklanmıştır. Resim tabanlı steganaliz yöntemi ile ses tabanlı steganaliz yöntemleri bu çalışmada detaylı incelenmiş ve bu ses ve resim tabanlı steganografi çözümlerin sağlamlığı bu çalışmada geliştirilen yazılımlar ile test edilmiştir. Bunun yanında, steganalizde kullanılan resim steganaliz için önerilen ve kullanılan yazılımlar ile bu yazılımların geliştirilmiş durumlarının gerçeklenmesi sonucunda elde edilen sonuçlar karşılaştırılmıştır. Yazılımların testleri için Ki-kare saldırısı kullanılmış ve sonuçlar değerlendirilmiştir. Elde edilen sonuçlarda resim tabanlı steganaliz kullanıldığında yedi resimde iyi sonuç elde edilmiştir. Fakat gürültülü resimlerde Ki-kare saldırısı yetersiz kalmıştır. Ses tabanlı steganalizde geliştirilen yöntemde ise çok başarılı sonuçlar elde edilmiştir. Ancak, resim steganalizde olduğu gibi gürültülü seslerde de iyi sonuçlar elde edilememiştir.

Bilim Kodu : **912.1.011**
Anahtar Kelimeler : **Steganaliz, Steganografi, FilpEmbed, Ki-Kare Saldırısı, Çift Değerler, Bilgi Gizleme, Son Bite Ekleme Yöntemi, Ses**
Sayfa Adedi : **237**
Tez Yöneticisi : **Prof.Dr. Şeref SAĞIROĞLU**

COMPARISON FOR STEGANALYSIS APPROACHES
(M.Sc. Thesis)

Muhammad D.HASSAN

GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
October 2008

ABSTRACT

Developing of technology today brings to ensure the security of data in the digital media with it. Coding and steganography techniques include the solution techniques which are used to ensure this security. The soundness of those techniques are also tested with the analysis techniques. In this study, stag-analysis methods, which are used to reveal the hidden data within the picture files and sound files, have been examined and the results, which are obtained from those methods, have been compared. In addition, steganalysis techniques, software and approaches and chi-square technique have examined and explained in detail in this study. Picture based steganalysis method and sound based steganalysis methods have been examined in detail in this study and the soundness of those picture and sound based steganography solutions has been tested via software which was developed in this study. In addition to this, the picture, which is used in the steganalysis, software, which is recommended and used for the steganalysis, and outcomes, which are obtained as a result of realization of the cases that are developed for this software, have been compared. Chi-square attacking has been used for the testing of software and the outcomes have been assessed. In the obtained outcomes, when the picture based steganalysis has been used, the good outcomes have been obtained in seven pictures. However, in the noisy pictures, Chi-square attack was insufficient. Very successful outcomes have been obtained in the method which was developed in the sound based steganalysis. However, as being in the picture

steganalysis, no good outcomes have been obtained for the sounds which included the prediction in the noisy sounds.

Science Code : 912.1.011
Key Words : Steganalysis, Steganography, FilpEmbed, Chi-Squared Attack, Data Hide, Least Significant Bit, Image, Sound
Page Number : 237
Adviser : Prof.Dr. Şeref SAĞIROĞLU

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren ve kıymetli tecrübelerinden faydalandığım Sayın Hocam Prof. Dr. Şeref SAĞIROĞLU'na şükranlarımı sunarım.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
LSB	(Least Significant Bit), En önemsiz bit
ANKTS	(Artificial Neural Network Technology for Steganalysis) Steganaliz için ileri yapay sinir ağları teknolojisi
A4	A4 Kağıdı
EAS	(Electronic Article Surveillance) Elektronik Makale Gözetleme
CIA	(Central Intelligence Agency), Merkezi Haberalam Teşkilatı
ASCII	(American Standard Code for Information Interchange), Bilgi değiş tokuşu için Amerikalı standart kod
JPG	(Joint Photographic Group)
JPEG	Joint Photographic Experts Group), Fotoğrafik uzman grubunu bitiştir
GIF	(Graphics Interchange Format), Grafik değiş tokuş biçimi
PNG	(Portable Network Graphics Format), Taşınabilir ağ grafikleri, biçimler
IR	(Infrared), Kızıl Ötesi
DNA	Deoxyribonucleic Acid
HTML	(Hyper Text Markup Language)
DP	Desibel
MP3	MPEG-1 Audio Layer-3
WEP	(Wired Equivalent Privacy), Kablolu çekilen eşit gizlilik
WAV	(Waveform), Dalga biçimi

Simgeler	Açıklama
AVI	**(**Audio Video Interleave), Ses görüntüsü, sırayla birleştirir
AU	(Audio Unit), Ses aygıtı
MPEG	(Moving Picture Experts Group), Film uzmanları, gruplar
BMP	(Bitmap), Bit Eşlemi
PNG	(Portable Network Graphics), Taşınabilir ağ grafikleri
TCP	(Transmission Control Protocol), Transfer kontrol protokolü
TV	(Televisions), televizyon
ISDN	(Integrated Services Digital Network), Bütünleşmiş, dijital şebekeye bakar
GSM	(Global System for Mobile), Mobil için evrensel sistem
RGB	(Red-Green-Blue), Kırmızı-Yeşil-Mavi
YCbCr	**(**Luma blue-difference, red-difference),
HSV	(Hue, Saturation, Value), Renk tonu, Doyum, Değer
HSL	(Hue, Saturation, Lightness), Renk tonu, Doyum, Açıklık
USAF	(United States Air Force), Birleşik Devletler Hava Kuvvetleri
MSE	(Mean Squared Error), Ortalama karesel hata
RMSE	(Root Mean Squared Error), Kök, ortalama karesel hatadır
PSNR	(Peak Signal to Noise Ratio)
DÇ	(Pairs of Values), Değer Çiftleri
KB	(Kilo Byte), Kilo Byte
RS	(Regular, Singular), Düzenli, tekil
RQP	(Raw Quick Pairs), Çiğ çabuk çiftler
DCT	(Discrete Cosine Transform), Ayrık Kosinüs Dönüşümü

Simgeler	Açıklama
DFT	(Discrete Fourier Transform), Ayrık Fourier Dönüşümü
FLD	(Fisher's Linear Discriminant)
DRM	(Digital Rights Management), Dijital hak yönetimi
CDN	(Content Distribution network)
USA	(United States of America), Amerika Birleşik Devletleri
VB	Ve Benzeri
StegAnalyzer	Steganography Analyzer
AS	**(**Artifact Scanner), Yapay Olgu Tarayıcı
SS	**(**Signature Scanner), İmza Tarayıcı
SRT	**(**Real-Time Scanner), Gerçek-Zaman Tarayıcı
DCCI	(Defense Cyber Crime Institute), Savunma siber suç enstitüsü
CSL	(Cyber Science Laboratory), Siber bilim laboratuvarı
FE	(Flip Embeded), FE aracı
PNM	(Portable Any Map), Taşınabilir Herhangi bir Harita
TXT	Metin dosyası
XML	(eXtensible Markup Language)
PCX	(PC Paintbrush Graphic)
AIFF	(Audio Interchange File Format), Ses Değiş Tokuş Dosya Biçimi
PDF	(Portable Network Graphic), Taşınabilir Şebeke Grafiği
MIDI	(Musical Instrument Digital Interface), Müzik Aleti Dijital Arayüz
EXE	(Executable), Çalıştırılabilir dosya

Simgeler	Açıklama
DLL	(Dynamic Link Library), Dinamik Bağlantı Kütüphanesi
DOC	(Document), Belge formatı
TIFF	(Tagged Image Format), Etiketlendirilmiş Resim Formatı

1.GİRİŞ

Bilgi gizleme yönteminin önemli bir alt disiplini olan Steganografi, bir nesnenin içerisine bir verinin gizlenmesi olarak tanımlanabilir [1]. Steganografi kelimesi kökleri "στεγαυος" ve "γραΦειν"'den gelen Yunan alfabesinden türetilmiştir. Tam olarak anlamı "gizlenmiş yazı","kaplanmış yazı" (covered writing) demektir [2].

Steganografinin gelişmesine paralel olarak gelişen diğer biri bilim dalı ise steganalizdir. Steganalizin amacı ise steganografinin tam tersine bir resim, ses veya herhangi bir dosyadaki gizli verinin varlığını ortaya çıkarmaktır. Steganaliz taşıyıcı, orijinal objelerle stego objeler arasındaki ayrımı yapmak üzere tasarlanmış teknikler bütünüdür.

Steganaliz çalışmalarının temelinde saklanan verinin taşıyıcı nesne üzerinde bir takım parmak izleri bıraktığı düşüncesidir. Yani saklama işlemi sonrasında oluşan stego nesne görsel, işitsel veya işlevsel olarak orijinalinden ayırt edilemez olsa da istatistiksel olarak belirgin farklılıklar taşımaktadır [3].

Steganaliz konusunda özellikle resim ve ses dosyalarının steganalizinde 2000'li yıllarda birçok çalışma yapılmıştır [3].

Bir steganografi uygulamasının güvenliği, stego nesnelerin ayırt edilebilir farklılıklar içermemesiyle ölçülmektedir. Eğer stego nesne içerisinde veri saklı olduğunu rastgele yapılacak tahminlerden daha iyi oranla tahmin eden bir algoritma mevcutsa o Steganografi sistemi kırılmış anlamına gelmektedir.

Gizli mesajın varlığının tespit edilmesi, mesajın uzunluğunu tahmin edilmesi ve gizli verinin tamamen ortaya çıkarılması konuları steganalizin sahasına girmektedir.

Steganografik uygulamalar karşı geliştirilen steganaliz ataklarından bazıları şunlardır [4]:

- Stego dosyanın bilindiği atak (stego-only)
- Taşıyıcı dosya ile stego dosyanın ikisinin de bilindiği atak (known cover)
- Saklı bir mesajın bilindiği atak (known message)
- Algoritma ve stego dosyanın bilindiği atak (chosen stego)
- Algoritma ve saklı bir mesajın bilindiği atak (chosen message)

Steganografide kullanılan saklama algoritmasına ihtiyaç duyulmaksızın evrensel tespit yapabilen steganaliz çalışmalarının sayısı artmaktadır. ANKTS (Artificial Neural Network Technology for Steganalysis) isimli projede de yapay sinir ağları kullanılarak dijital resimler istatistiksel olarak analiz edilip saklı bir veri olup olmadığının evrensel olarak tespit edilmesine çalışılmaktadır [5].

Görüntü dosyaları üzerinde bilgi gizlemek için çeşitli steganografik yöntemler geliştirilmiştir. Bunlar 3 başlık altında sınıflandırılabilmektedir [6]:

- En önemsiz bite ekleme
- Maskeleme ve filtreleme
- Algoritmalar ve dönüşümler.

En önemsiz bite ekleme en yaygın kullanılan bilgi gizleme yöntemlerinden biridir. Taşıyıcı ortamın en az önemli bitlerini insan gözünün fark edemeyeceği şekilde gizli veriyi saklamak amacıyla değiştirmeyi temel alır.

Maskeleme ve filtreleme yöntemleri genellikle 24 bit resimler için kullanılmakta olup resmin en önemsiz alanlarının tespit edilerek buralarda saklama yapılmasını temel almaktadır. Bu yöntemler genelde filigran uygulamalarında karşımıza çıkmaktadır. Maskeleme teknikleri JPEG formatındaki resim dosyaları için daha uygundur.

Dönüşümler ise yine daha çok JPEG dosyalar üzerinde kullanılmaktadır. En yaygın olarak kullanılan dönüşümler ise DCT (Discrete Cosine Transform) ve DFT (Discrete Fourier Transform)'dır.

Şu anda internette herkese açık çok fazla ve basit Steganografi uygulamaları olmasına rağmen, güvenilir tespit yapabilen steganaliz sistemleri çoğunlukla mevcut değildir. Gelecek günlerde bu ve buna benzer çalışmalarda yoğun artışlar olacağı değerlendirilmektedir.

Takip eden bölümlerde resim ve ses steganalizi konusunda yapılan çalışmalar ve geliştirilen yöntemler daha detaylı bir şekilde anlatılmaktadır.

Bu tez çalışması takip eden şekilde yapılandırılmıştır. İkinci bölümde, Steganografi konuları genel bir şekilde incelenmiştir. Steganografinin tarihi, geçmişteki ve günümüzdeki Steganografi uygulamaları ve bu uygulamaların metin, ses, film, hareketsiz görüntüler ve diğer Steganografi biçimleri üzerinde gerçekleştirilmiştir. Ayrıca beş steganografik yaklaşım açıklanmıştır. Steganografik sistemi değerlendirme kriterleri (taşıyadaki değişim, kapasite ve dayanıklılık) açıklanmıştır.

Üçüncü Bölümde, resim steganalizi tanımlanmakta, (Görsel Saldırılar, Değer Çiftlerinin İstatistiksel Analizi , İkili İstatistik Metotları, JPEG Uyumluluğunu Esas Alan Steganaliz, Sayısallaştırma Matrisinin Türetilmesi, Evrensel Kör Steganalizi, Benzersiz Parmak İzleri ve RQP Steganaliz) yedi kategoriye sahip olan resim steganalizi sunulmuş, hangi yöntem diğer yöntemlerden üstün olduğu karşılaştırılmış ve her yöntem için bir örnek uygulama verilmiştir.

Dördüncü bölümde, Steganaliz yazılımlarını ele almıştır. Her yazılımın tanıtımı, kullanışı ve incelemesinden bahis edilmiştir.

Beşinci bölümde, 8 bitlik gri seviyesindeki resimlere ki-kare saldırısı gerçekleşmiş ve sonuçlar analiz edilmiştir. Bunun yanında DÇ algoritması detaylı bir şekilde sunulmuş, kararlar ve varsayımlar açıklanmıştır.

Altıncı bölümde, 8 ve 16 bitlik iki tür dosya uzantısına (Wav,Au) ki-kare saldırısı gerçekleşmiş ve kararların yanı sıra varsayımlar da detaylı bir şekilde açıklanmıştır. Bunların analizini ve saldırının başarısına dair detaylar sunulmuştur.

Sonuç bölümde ise, yapılan çalışmalar ışığında, bu tez kapsamında yapılan incelemer, deneyler ve elde edilen sonuçlar kapsamlı olarak değerlendirilmiştir.

2. STEGANOGRAFİ

Steganografi, eski Yunancadan dünya dillerine geçmiş olan bir terimdir. "*Stegano* = gizli, saklı" ve "*Grapy* = yazmak, çizmek" kelimelerinden oluşan bu kelime, kabaca; "*gizli yazı*" olarak dilimize çevrilebilir. Lau Stephan'ne göre steganografi, "görünüşte zararsız olan haberleşmenin içine gizli haber ilave ederek haberleşmektir" [42].

Steganografide amaç, mesajın varlığını gizlemektir. Bunun için, görünmez mürekkep kullanımı, mikronoktalama (microdot), metin düzenleme, boş (null) şifreleme gibi teknikler kullanılır. Bu açıklamalardan da anlaşılacağı gibi steganografi, sayısal olsun ya da olmasın "hedefe ulaşması istenen, gizliliğe sahip herhangi bir düz metnin, iletilmesinin / alınmasının, üçüncü şahısların haberi olmayacak şekilde, gizli olmayan medya vasıtasıyla gerçekleştirilmesine" denir.

Steganografi yaklaşık 2500 yıldır, değişik biçimlerde, özellikle askeri, diplomatik, kişisel ve telif hakkı uygulamalarında yaygın olarak kullanılmaktadır. Kısacası, steganografi saklanan mesajın herhangi bir gözlemcinin fark edemeyeceği bir nesnenin içerisine bir mesajı gizlemek için uygulanan işlemler bütünü olarak tanımlanabilir.

Steganografi, herkes tarafından görünmeyen bir iletişim çeşidi olup, gizli yazım sanatıdır ve bir mesajı gizleyerek onun anlaşılabilirliğini ve çözülebilirliğini engelleyerek iki taraf arasında güvenli bir iletişim sağlamayı amaçlar [10].

Steganografi aslında şifreleme biliminin (kriptografi) alternatifi değil onun tamamlayıcısıdır. Kriptografide, üçüncü kişiler gizli bir iletişimin varlığını bilirler. Ancak, alıcı ve vericinin en temel varsayımı, üçüncü kişinin takip edebildiği gizli iletişimin içeriğini "makul" bir sürede çözememesidir. Oysa steganografide gizli iletişimin varlığı saklanmaktadır [96].

Günümüzde steganografi, teknikleri güvenliği daha da arttırmak için şifrelenmiş

verileri gizlemek için genelde ses, sayısal resim veya görüntü (video) dosyalarını kullanmaktadır. Şifrelenmiş veriler kendi başlarına saldırganların (hacker) dikkatini çekerken, görüntü ya da ses dosyalarının içine gizlenmiş olduklarında hiç kimse fark etmeyeceğinden kırılmaya da çalışılmayacaktır. Steganografi ile ilgili daha detaylı bilgiler aşağıda farklı başlıklar altında incelenmiştir.

2.1. Steganografi Tarihi

İran savaşları sırasında, Herodot, kafasını kazıtıp kafa derisinin üzerine, gizli bir mesajın dövmesinin yapılmasına izin veren bir ulaktan bahseder. Mesaj yazıldıktan sonra ulak saçının uzamasını bekler. Daha sonra mesajı bekleyen kişiye ulaşır ve kafasını tekrar tıraş eder. Mesaj ortaya çıkar. Bu, steganografinin tarihte ilk kullanımıdır [43].

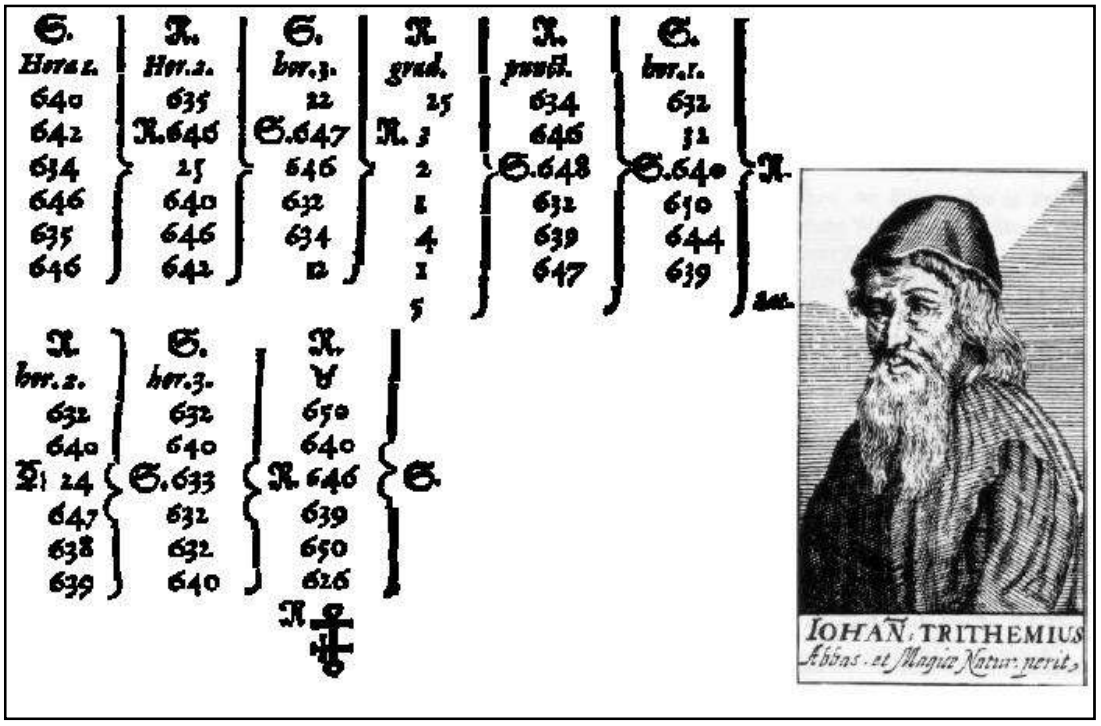
Antik Yunan'da tahtanın üzerine yazılı mesajlar balmumu ile kaplama kullanılmamış tablet ("wax tablet") olarak gizleme, avlanan hayvanların karnında mesaj gönderme ve görünmez mürekkeple gizli mesajlar yazma kullanılan yaklaşımlardan bazılarıdır. Bu mesajları okuyabilmek için kağıdı ısıtmak veya kağıda kimyasal bir işlem uygulamak gerekmektedir [43].

Steganografinin kullanımı yalnızca Yunanistan ile sınırlı değildir. Çinliler de kendi kaynaklarında meyve sepetini nasıl gizli iletişim için kullandıklarını anlatmaktadırlar. Meyve sepetindeki her meyvenin birbirine göre pozisyonu farklı bir anlam ifade edecektir [6].

2.2. Geçmişteki Steganografi Uygulamaları

Johannes Trithemius 1462–1516 yılları arasında yaşamış Alman bir rahiptir. Onun kriptoloji, astroloji ve sayı dizeleri ile ilgili kitabında" Steganografi: yazının gizlendiği ve tekrar geri elde etmenin insan zekası gerektiği bir sanat dalı" olarak tarif edilmiştir [27].

Gina Kolata [28] “Thomas Ernst ve Jim Reeds isimli iki araştırmacı, Şekil 2.1’de verlen kitabın gizli kodlar içerdiğine inanarak bir çalışma yaptılar. Ernst Pittsburgh Üniversitesinde öğrenci iken bu konuda 200 sayfalık almanca bir rapor yayınlamış, 1996’da Hollanda dergisi *Daphnis*’te yayımlanmış fakat çok az ilgi görülmüştür”. Reeds, AT&T laboratuarlarında bir matematikçi olarak çalışırken Trithemius’un bu kitabı üzerinde çalışmaya başlamıştır. Trithemius’un çalışmaları üzerine bilgi toplarken Ernst’ın raporunu ortaya çıkarmıştır.



Şekil 2.1. Johannes Trithemius‘un Steganografi ile ilgili 3.Kitabında yer alan sayı Çizelgelerinden bir örnek [27].

Bu iki araştırmacı kısa bir süre sonra, kitabın içerisinde gizli mesajların olduğunu keşfettiler. Çizelgelerin içerisinde yer alan gizli mesajlar çok az ilgi görmüştür. Mesajlardan bir tanesinde “Hızlı kahverengi tilki tembel köpeğin üzerinden zıpladı” denilmektedir. İkinci mesaj ise, “Bu mektubu taşıyan kişi dolandırıcı ve hırsızdır Kendini ona karşı koru. Sana bir şeyler yapmak istiyor”. Ve üçüncü mesaj ise İncilin 23.bölümünün birinci kısmını içermektedir [44].

İskoçya kraliçesi Mary Queen mektuplarını gizleyebilmek için kriptografi ve steganografi tekniklerini kullanmışlardır. Yazdığı mektupları bir bira fıçısının deliğinde gizlemiş ve kendisi o sayede hapishaneden kurtulmuştur [100].

Steganografinin diğer kullanımları, normal yazı araçları ile sınırlandırılamaz. Coğrafik taslaklar da steganografinin bir biçimi olarak kabul edilebilir. Şekiller, görünüme açıktır fakat birçok görüntü havadan bakılmadıkça tespit edilemez [24].

Aşağıda, ikinci dünya savaşında kullanılan bir steganografi örneği verilmiştir [112].

"Apparently neutrals protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretex for embargo on by-products, ejecting suets and vegetable oils."

Yukarıda verilen paragrafta her kelimenin ilk harfleri yan yana getirildiğinde*"Pershing sails from NY June 1."* mesajı ortaya çıkmaktadır.

Diğer bir steganografik yöntem, İkinci Dünya Savaşı sırasında Amerika Birleşik Devletleri Deniz Kuvvetleri tarafından Navajo şifre konuşucularıdır. Şifre konuşucular basit bir şifreleme tekniği kullanarak, mesajlar açık bir metin içerisinde gönderilmiştir.

Steganografinin bir diğer uygulaması ise Cardano ızgarasının kullanımıdır [29]. Bu alet, ismini yaratıcısı olan Girolama Cardano'dan almıştır ve üzerinde kesilmiş delikler olan basit bir kağıt parçası olarak düşünülebilir. Izgara basılmış bir metin üzerine yerleştirildiği zaman istenen mesaj elde edilebilmektedir. Cardano ızgarasına benzeyen teknikler içerisinde, klasik steganografi teknikleri metin üzerindeki pin deliklerini (örneğin gazetelerde) ve kalem ile tekrar yazılmış metinleri içermektedir.

Steganografik tekniklerden en son kullanılanı fotoğraf üzerindedir. Vietnam savaşı esnasında Birleşik Devletler ordusundaki esir alınan askerler çekilen fotoğraflarda değişik el kol hareketleri yapmışlar daha sonra bu fotoğraflardaki bu hareketler medya tarafından çözülmüştür [30].

Çizelge 2.1. Vietnam Savaşı sırasında esir düşen Amerikalı askerlerin kullandığı 5x5'lik kod

	1	2	3	4	5
1	A . .	B . ..	C, K	D	E . .....
2	F .. .	G	H	I	J .. ..
3	L	M	N	O	P ... .
4	Q	R	S	T	U ....
5	V	W	X	Y	Z ....

Diğer tekniklerden bir tanesi Mors alfabesine göre göz kapaklarını kapatıp açarak kelimeleri kullanmaktır. Kötü şöhretli bir hapishane olan Hanoi Hilton'daki mahkumlar kendi aralarında iletişim kurmak için "tıklatma kodu" kullanmaktadır. Kod, her bir harfin bir tıklatma sırasına sahip olduğu 5x5'lik bir matrise dayanmaktadır. Harfler arasındaki boşluklar varsa iki kez arka arkaya tıklatma yapılmaktadır.

2.3. Günümüzde Steganografi Uygulamaları

Günümüzde, dijital steganografinin çeşitli biçimleri üzerinde önemle durulmaktadır. Genellikle, metin dosyaları, hareketsiz görüntüler, film görüntüleri ve ses dosyaları gibi birçok sayısal teknolojileri kullanmaktadır. Birçok sayısız steganografik yöntem olmasına karşın, bunlar iki grup başlık altında değerlendirebilir [45].

Görüntü Alanı (Image Domain) aracı grubu, en az önemli bit (LSB) eklemesi ve gürültü temizleme işlemlerini uygulayan bit temelli yöntemleri kapsamaktadır. Dönüşüm Alanı (Transform domain) aracı grubu ise, Ayrık Kosinüs Dönüşümü (DCT) ve dalgacık dönüşümü (wavelet transformation) gibi görüntü dönüştürücüler ve algoritma işlemlerini içermektedir [45].

11 Eylül 2001 olaylarından sonra, steganografinin olası kullanımlarını tespit edebilmek için bütün ağ ortamları taranmıştır. Ayrıca internet üzerinde servis sunan bazı pornografik web sitelerindeki görüntüler üzerinde de steganografik mesajların olabileceği rapor edilmiştir. Bu rapora istinaden Michigan Üniversitesinden bir grup internet üzerinde bulunan eBuy – çevrimiçi satış gibi çeşitli siteler üzerinde bulunan iki milyonu aşkın resimleri taramışlardır. Fakat bu resimlerin incelemesinde şüpheli herhangi bir resim bulunmamıştır.

Görüntü üzerindeki gizli mesajlar kaygısının ötesinde, Bin Laden'in televizyon yayınlarını içeren görüntüler üzerinde de steganografik mesajlar olabileceği kaygısı yaşanmaktadır.

Steganografinin yalnızca sayısal teknolojinin bir özelliği olmadığı göz önünde bulundurularak, yayınların ses kısmı içerisinde gizli mesajların olabileceği hatta televizyon görüntülerinin arka planında gizlenmiş mesajların olabilme ihtimali vardır.

Herhangi bir organizasyon steganografik teknolojileri kullanarak bilgi gizlemesi yapıyor mu? Evet, bilgiyi gizleyebilmek için birçok organizasyon steganografik teknolojiyi kullanmaktadır. Web üzerindeki organizasyonlara ait web sitelerini araştırarak bu soruya yanıt bulmak amacıyla Ulster Loyalist Volunteer Force (www.ulisnet.com) sitesi tespit edilmiştir. Bu sitenin steganografi tabanlı bilgiler gönderdiği belirlenmiştir. Hazırlanan raporun ilk paragrafında " İdeal bir dünya içerisinde, şifrelenmiş e-postaları veya dosyaları birbirimize herhangi bir saldırı veya misilleme korkusu olmadan açıkça gönderebiliriz. Fakat bu durum gerçek hayatta ne yazık ki böyle değildir. Çünkü şifrelenmiş e-postaları izin vermeyen bir şirkette çalışıyor olabilirsiniz veya hükümet şifrelenmiş iletişime onay vermiyor olabilir. İşte steganografinin devreye girdiği nokta tam olarak burasıdır. Steganografi, bilginin bilgi içerisinde saklanılmasıdır." denilmektedir [13].

Steganografik teknikleri kullanan diğer organizasyonların büyük bir çoğunluğu bireylerin veya şirketlerin fikirsel ve zihinsel mülkiyetlerini koruyabilmek için bu

yöntemden faydalanmaktadır [13]. İş Yazılım Birliğine (Business Software Alliance) göre yazılım hırsızlığı veya lisanssız yazılım kullanılması bu problemin temelini oluşturmaktadır.

Business Software Alliance tarafından yayımlanan rapora göre [83], bir başka endüstri ticaret grubu yazılım endüstrisi zarar ve kayıplarının tüm dünya genelinde yıllık 11.4 milyon doların üzerinde olduğunu tespit etmiştir. Aynı grubun Amerika Birleşik Devletlerindeki bütün yazılımların %25'den fazlasının kaçak ve kanunsuz bir şekilde kopyalandığının bildirmesine karşın, Güney Asya ve Doğu Avrupa gibi kıtalarda yer alan ülkelerde durum oldukça vahimdir. Buradaki uzmanların söylediğine göre bütün uygulamaların yaklaşık %95'i korsan şeklinde kullanılmaktadır.

Diğer bir kullanıcı grubu ise, yaygın olarak steganografik teknikleri sayısal damgalama teknikleri uygulamalarının içerisinde kullanmaktadır [92]. Birçok teknik kullanılarak, görüntüler, müzikler, filmler sayısal damgalama ile birlikte üretilebilir. Sayısal damgalamalar değişik birçok konfigürasyon içerisinde yer alabilmektedir; Kırılgan-sağlam, görünür-görünmeyen, özel-halka açık gibi. Kırılgan damgalamalar kolaylıkla görüntü işleme ile tahrip edilebilir ve görüntü yetkilendirme (authentication) sistemleri içinde kullanımı bulunmaktadır [92].

Steganografik yaklaşımları kullanan aşağıdaki gibi birçok ticari yazılım uygulamaları, servisleri ve firmaları mevcuttur; Alpha-Tec, Cognicity, Giovanni, Digmarc, Signum Technologies, SysCoP ve Verance Digital bunlardan sadece birkaç tanesidir [25]. Test tarafında ise StirMark ve unZign sayısal damgalamanın sağlamlığını ve etkinliğini test etmeye yönelik ücretsiz (freeware) programlar sunmaktadır.

Sayısal damgalama, sadece sayısal ortam ile düşünülüp sınırlandırılmamalıdır. Örneğin, Xerox firması yazdırılan görüntüler üzerinde çalışabilen yeni bir sayısal damgalama sistemi geliştirmiştir. Bu teknik, bir anahtar görüntü ile birlikte (half tone) yarım ton farklı bir görüntü üretmeyi gerektirmektedir. Bu ekran, ince bir baskı

ile orijinal görüntünün tam üzerine yerleştiği zaman anahtar görüntü gözükecektir. Bu teknik, görüntünün önemli bir ölçüde azaltılmasıyla zorlaşmakta ve anahtar görüntünün kaybolmasıyla sonuçlanmaktadır [92].

Stokastik ekranlar normalde yarım ton analog ekranlardır ve yazma işleminde sıklıkla kullanılırlar fakat noktalar stokastik ekranların düzenli bir şekilde yerleşmemesini sağladığından genellikle oldukça küçüktür (tipik olarak 21 mikron düzeyinde noktalar). Stokastik ekran, ekran bir firmware olarak kabul edilmek koşuluyla sayısal görüntüler üzerinde kullanılan algoritmalara benzetilebilir, böylelikle bu tekniğin sağlamlığı arttırılır.

(a) (b)

Şekil 2.2. Lena'nın Resmi [31]
a) Orijinal resim
b) damgayı ortaya çıkarmak için stokastik ekran ile birleştirilmiş görüntü

Steganografi ve sayısal damgalama bilimi geliştikçe, saldırı ve atak yöntemlerinin çeşitliliğinde de o denli bir ilerleme görülmektedir. Saldırılar 20 kategoride ifade edilir [99]:

- İlave Gürültüler (Additive Noise)
- Filtreleme (Filtering)
- Kırpmak (Cropping)
- Sıkıştırma (Compression)
- Döndürme ve Ölçekleme (Rotation and Scaling)

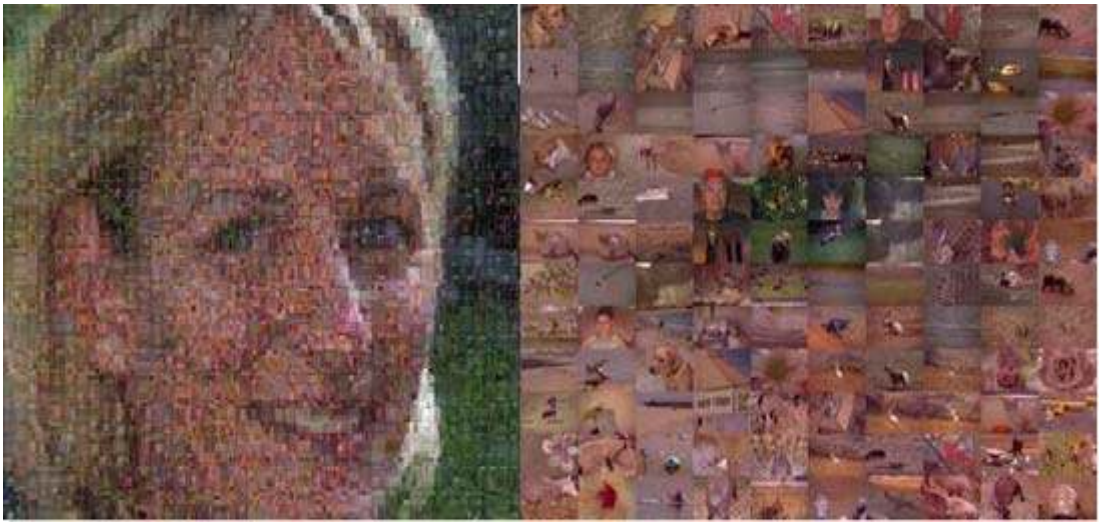
- İstatistiksel Ortalama (Statistical Averaging)
- Çeşitli damgalamalar (Multiple Watermarking)
- Diğer seviyedeki ataklar

Sayısal steganografi günümüzde yaygın olarak kullanılmasına rağmen, steganografinin diğer biçimleri özellikle fiziksel güvenlik tedbirlerinde halen kullanılmaktadır. Nazilerin öncülük ettiği küçük resimlerin kullanımı, bir aracı işaret etmek için basitçe kullanılabilecek bir yöntem olarak kabul edilir. Bu küçük resimlerle Gösterilen araç ve gereçler hırsızlık durumuna karşı tanımlanmaya yardımcı olurlar. Küçük resimleri veya fotoğrafları bu amaçla kullanan birçok firma bulunmaktadır [25]. Makale yazarları Elektronik Makale Gözetimi (Electronic Article Surveillance - EAS) etiketlerinin de bir çeşit steganografik form olduğunu iddia etmektedir. Bu etiketler belirgindir ve etiket içerisindeki mesaj silinmediği müddetçe, kim geçmiş depolanmış makale tarama mekanizmalarını kullanarak bir makale ararsa, ona kesin bir mesaj vermektedir. Bir başka deyişle "Bu makale henüz etkinliği kaldırılmamıştır ve eklenen makale çalınmış olabilir".

Şekil 2.3. RF EAS etiketinin gizli mesaj tarafı. "Bizimle alışveriş yaptığınız için teşekkürler" yazılı bir mesaj içermektedir [109].

Steganografik örnekleri incelerken birkaç ayrıntı da yüzeysel bir göz atmayı gerektirmektedir. Örneğin, fonograf ses kayıtları üzerinde gizlenen mesajlar. Değişik ses kayıtları geriye doğru çalındığında farklı mesajlar özellikle satanist mesajlar içerebildiği konusunda birçok rapor mevcuttur. Bu konudaki iddiaların birçoğu kanıtlanamasa da bugünün teknolojisiyle bu basit tekniği kullanarak mesaj üretilme ihtimali her zaman vardır.

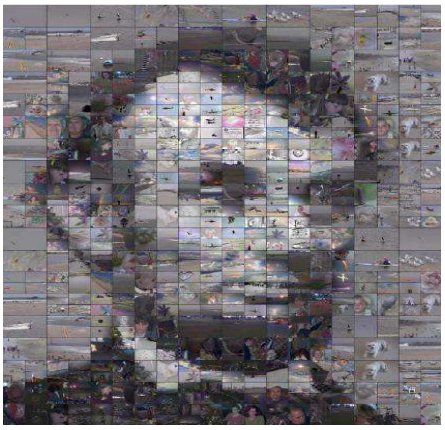
Bir diğer steganografik yöntem ise bilinçaltı önermelerin kullanımını içermektedir. 1950'li yıllarda, Amerika halkı tiyatro sahnelerinde gösterilen bilinçaltı mesajlarından oldukça etkilenmiştir ve bu konuda Central Intelligence Agency (CIA) tarafından önemli araştırmalar yapılmıştır [32]. Bilinçaltı önermeleri reklamlardan (değişik gruplardan alınan görüntüler ve mesajlar farklı bir anlam içerebilir), InnerTalk gibi modern bilinçaltı önerme programlarına kadar birçok alanda çeşitlilik gösterir. Steganografik bir yöntem olarak sınıflandırılmamasına rağmen, PhotoTiled resimler içerisinde bilgiyi gizlemek mümkündür. Resimler üzerinden görüntüler yaratan bir program, William Leigh Hunt tarafından hazırlanan bir web sitesi mevcuttur [73];



(a) (b)

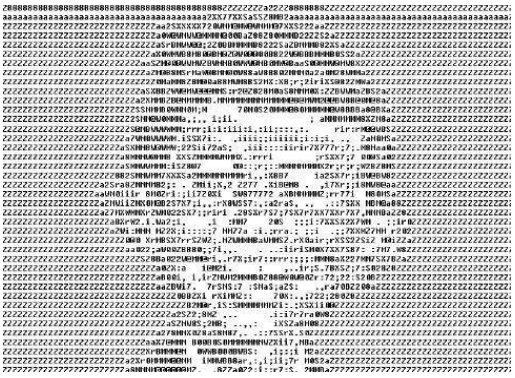
Şekil 2.4. PhotoTiled yöntemi ile bir görüntü
a) PhotoTiled görüntü.
b) sol gözün genişletilmiş görüntüsü.

Bu uygulama, tamamen yeni bir resim yaratmak için fotoğraf koleksiyonlarının işlemesine ilişkin çarpıcı örnekler sağlamasına karşın, tartışılması gereken konu çıkış görüntüsünde her zaman gizli bir bilgi içerebilme ihtimali olduğudur. Uygulamanın kendisi, PhotoTile yüksek oranda denenmiş ve sık kullanılan bir paylaşımlı yazılımdır (shareware programing) [74].



Şekil 2.5. Lincoln'un PhotoTiled yöntemi ile görüntüsü

Hafifçe farklı yönler kullanarak, görüntü dosyalarını "ASCII art" formatına dönüştüren uygulamalarda potansiyel olarak steganografik uygulamalar olarak kabul edilebilir. Renkli bir görüntü dosyasının ASCII metin görüntüsüne dönüşmüş biçiminin örneği aşağıdaki gibidir.



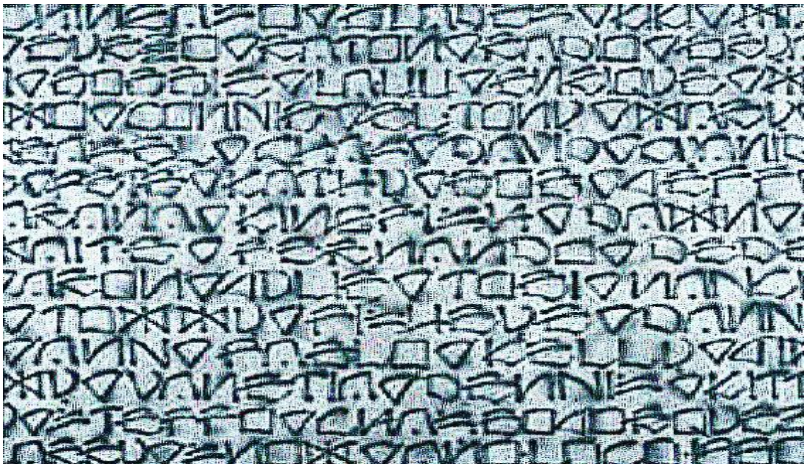
Şekil 2.6. Renkli bir resim görüntüsünden elde edilen ASCII metin dosyası

ASCII metin dosyasını görüntülendiğinde (uygun olduğundan burada jpg'e çevrilmiştir), metin dosyası kendi başına gizli bir dosya olmamasına rağmen, çalışkan bir birey görünen metin dosyası içerisine belirli harfler gizleyebilir.

Bir diğer sayısal format animasyonlu GIF formatıdır. GIF formatının normal steganalizi, bu biçimi kullanarak bir mesajı gizlemenin önemsiz olduğu durumda mutlak suretle gizli mesajları tespit edemeyebilir.

Steganografi konusunda, mesajları gizlemeye yönelik kullanılan diğer alanlar şunlardır [26];

• Holography (Üç boyutlu resim üretebilen fotoğraflama yönteme) teknolojisi
• Kızılötesi - Infrared (Özellikle bilgisayarlar için programlanabilir IR el kumandası)
• Çağrı cihazları
• Gizlenen mesajları görünebilir yapmak için dalga boylarını filtreleyen renkli gözlükler
• Mürekkep, manyetik, termokromik, fotokromik
• DNA mesaj saklaması
• Jargon (özel mesleki dil) konuşması
• Sabit disk sürücüleri, diğer disk sürücüler üzerindeki boş alanlar
• HTML kodları



Şekil 2.7. Charles Chaz Bojorquez'in eseri. (Placa, Roll) [110].

Şekil 2.7'de, yazar değiştirilmiş bir alfabe kullanmış ve kendi noktalama işaretlerini yaratmıştır. Bu sonsuz listenin hedef noktası mümkün olan steganografik taşıyıcıları listelemektir [110].

Önemli bir husus steganografik içeriğin internet üzerinde yer alan JPEG veya GIF görüntülerinden çok daha fazla bulunmasıdır. Indiana Üniversitesinde Yenilikçi Bilgisayar Uygulamaları Merkezinin yaptığı bir araştırmaya göre ortak olarak 141

adet görüntü dosya biçimi mevcuttur [105]. Bu aşamada, en fazla bulunabilen ve yaygın olarak kullanılan steganografik yöntemlere göz atmak yararlı olacaktır.

Sağıroğlu ve Tunçkanat (2003), gri seviyeli Bitmap resimleri içerisine, görsel olarak fark edilmeksizin, en önemsiz 4.bit seviyesine kadar, LSB modifikasyonu yöntemiyle veri saklanabileceğini gösteren Türkçe bir yazılım geliştirmişlerdir [98].

2.3.1. Metinsel steganografi

Metinsel steganografi içerisinde çok sayıda yöntem kullanılmaktadır [46];

- Açık alan uygulamaları (Open space methods)
- Satır kaydırmalı kodlama (Line shift coding)
- Kelime kaydırmalı kodlama (Word shifting coding)
- Sözdizimi ile ilgili yöntemler (Syntactic methods)
- Anlamsal yöntemler (Semantic methods)
- Öz nitelik kodlaması (Feature coding)

Metin içerisine beyaz boşluklar dikkate alınarak veri saklanabilmektedir. Bu şekilde 3 ayrı yöntem kullanılmaktadır. Birinci yöntemde her sonlandırıcı karakterin sonuna bir veya iki boşluk bırakarak kodlama gerçekleştirilir. İkinci yöntemde ise satır sonlarına boşluk bırakılarak kodlama yapılmakta ve veri saklanmaktadır. Üçüncü yöntemde de metinin sağa yaslanmasıyla oluşan boşlukların kodlanmasıyla veri saklama işlemi gerçekleştirilmektedir [65].

Metin içerisine veri saklama yöntemleri bunlarla sınırlı değildir. Örneğin bir metindeki kelimeler, ilk harfleri yan yana geldiğinde anlamlı bir mesaj oluşturacak şekilde sıralanarak da mesaj saklanabilir. Buda 2.Dünya Savaşında gerçeklenmiş örneklerden biridir.

Metin içerisine veri saklamayla ilgili ilginç bir uygulama “spam mimic” isimli bir web sitesi [76] tarafından gerçekleştirilmektedir. Bu sitede istenilen mesaj

steganografik yöntemlerle kodlanarak spam e-posta metnine dönüştürülmektedir. Steganografik uygulamalarda spam e-postaların potansiyel bir taşıyıcı olduğu değerlendirilmektedir [77].

Metin içerisine, metnin oluşturulduğu dilin söz dizim (syntax) yapısında ve eşanlamlı kelimelerin varlığından yararlanarak da veri gizlenebilmektedir [65]. Türkçede özne, nesne ve tümleçlerin cümledeki sırası anlamı değiştirmeden sıralanabilir ve farklı sıralama kombinasyonları kodlanarak, bu özellik veri saklamada kullanılabilir. Örneğin, "*Hayata bağlan Türkcell'le*" cümlesindeki kelimelerin 3! Yani 6 farklı sıralanışı vardır ve vurgu farklılığı olsa da sıralanışlar aynı anlamı vermektedir. Her bir sıralanış kodlanarak veri gizleme imkanı oluşturulabilir. Eşanlamlı kelimelerde de, kelimelerden birinin 0 diğerinin 1 değerini ifade etmesi kodlanarak metin içerisine veri saklanabilir. Örneklerden de anlaşıldığı gibi söz dizim ve eşanlamlılık durumlarının kodlanmasıyla metin içerisine veri saklanabilmektedir. Ancak saklama kapasitesinin çok küçük değerlerde olacağını da belirtmekte fayda vardır.

2.3.2. Ses / Film dosyası steganografisi

Mp3 dosyaları kullanımı giderek yaygınlaşan sıkıştırılmış ses dosyalarından birisidir. Boyutunun küçük olabilmesi nedeniyle steganografik uygulamalar için uygun bir taşıyıcı dosya türüdür. Mp3 dosyaları içerisine veri saklayabilen az sayıda uygulama mevcuttur. Bunların en bilineni Mp3Stego yazılımıdır. Bu yazılım sıkıştırma esnasında Mp3 dosyaları içerisine veri saklayabilmektedir [75].

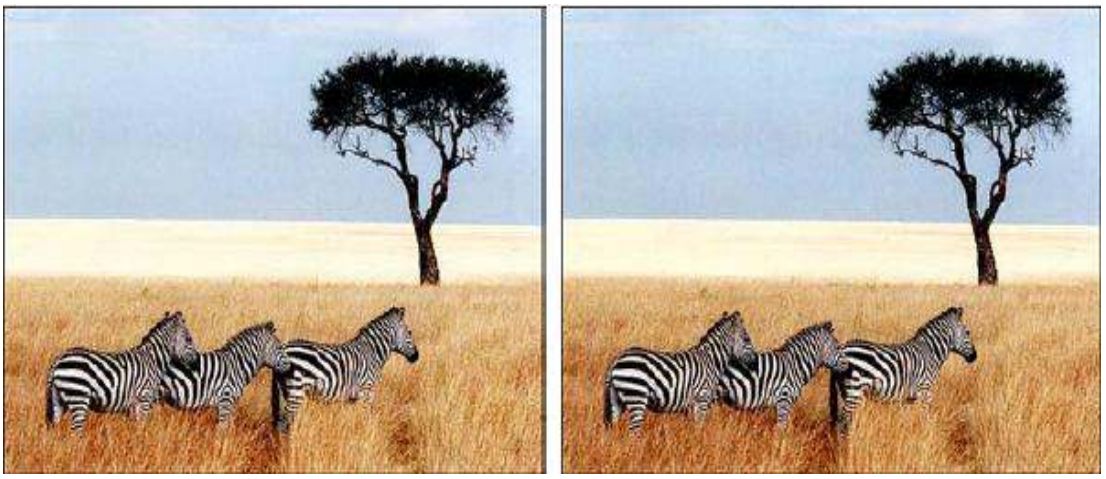
.wav, .avi, .au, ve mpeg formatları gibi çeşitli formatlarda gizlenmiş bilgileri öneren çeşitli uygulamalar bulunmaktadır. Bu uygulamalar verileri saklama, şifreleme, dosyalama işlemlerinden ibarettir ve bunlar kamuflaj / gizleme yapar gibi dosyanın sonuna gizlenmiş bilgileri ekler.

Adli ve Nakao (2005), internette sıkça kullanılan küçük boyutlu midi ses dosyaları içerisine, LSB modifikasyonu, tekrarlanan komut kodları algoritması ve sistem harici kodları algoritmasını kullanarak veri saklayan bir çalışma sunmuşlardır [101]. MIDI

dosyaları çok popüler ve küçük boyutlu olduğundan steganografik uygulamalarda kullanılması iyi bir seçim olabileceği vurgulanmıştır.

2.3.3. Hareketsiz görüntü steganografisi

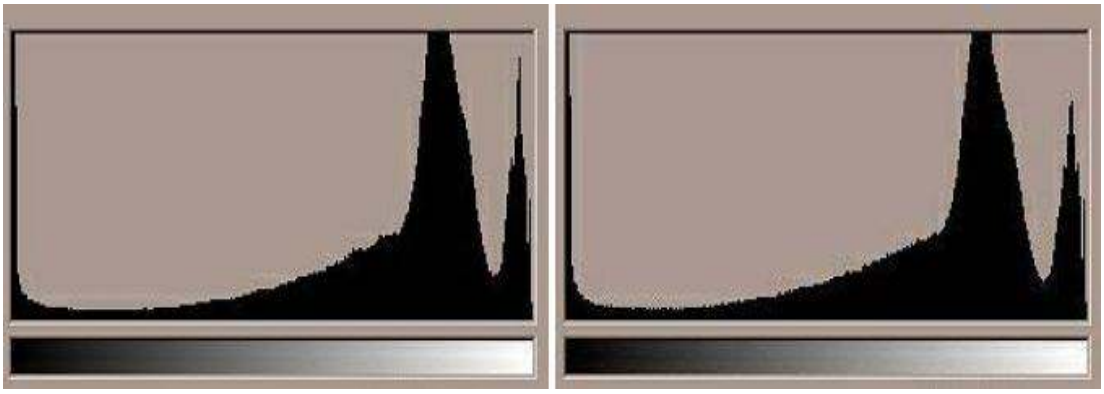
Hareketsiz görüntü steganografi konusu ile ilgili oldukça çok sayıda uygulama vardır. Andrew S. Tanenbaum (Minix İşletme Sistemlerinin Yazarı) bu konu ile ilgili ilginç ve etkileyici bir demo geliştirmiştir [78]. Demo S-Tools v4'in bir kopyasını sağlamaktadır. 1024x768 piksel boyutlarında iki BMP görüntüsü üzerinde çalışmıştır. Şekil 2.8'de verilen orijinal fotoğraf Andrew S. Tannenbaum'e aittir).



(a) (b)

Şekil 2.8. Tanenbaum demosundan Bitmap (.BMP) dosyaları.
a) orijinal görüntü.
b) gizli mesaj içeren görüntü.

Gözle görülebilir bir açıdan, her iki görüntüde mantıklı olduğu görünmektedir. Dosya boyutları yalnızca 2 bayt farklıdır, orijinal dosya için 2.359.352 ve gizlenen metindeki dosya için 2.359.350 dır. İmgelerin onaltılık sistem (hexadecimal) durumları oldukça farklı iken, her iki dosyanın histogramları fark edilemez şekilde aynıdır. Histogramların tüm sağ kısmında aşikar değişiklikler vardır. Bir test tabakası olarak (1024*768 piksel) oluşturulmuştur. Buna ek olarak, web'den elde edilen ikil eşlemli görüntü test edilmiştir [47].



(a) (b)

Şekil 2.9. Zebra görüntüsünün histogramı.
a) orijinal görüntünün histogramı.
b) steganografik görüntünün histogramı

Şekil 2.8'deki görüntü aşağıdaki gizli metinleri içermektedir;

• Hamlet 201,788 Byte
• Julius_Caesar 115,805 Byte
• King_Lear 176,952 Byte
• Macbeth 103,609 Byte
• Merchant 120,927 Byte
• Notice 15,810 Byte

Toplamda 734,891 Byte gizlenmiş veri orijinal görüntü üzerinde mevcuttur. Metinsel steganografi testinden sonra WbStego programına geri dönerek, sadece beyaz bir görüntü (1024 x 768 piksellik) test ortamı için yaratılmıştır. İlave olarak web'ten alınan bir görüntü test edilmiştir. 114 kelimelik gömülü kısa mesajdan sonra, görüntünün ne içindeki nesnelerde nede dosya boyutlarındaki değişikliğe sebebiyet vermediği görünmüştür. Görüntünün içindeki herhangi bir hatanın olduğuna dair bilgi görüntülerin hex dump'ını almıştır. Bu görüntüleri karşılaştırmak için kullanılan hex aracı "frhed, the free hex editor" adındadır ve internetten elde edilebilir [79].

Resmi, JPEG görüntüsüne dönüştürdükten sonra, benzer bir test beyaz iki eşlemli görüntüler üzerinde yapıldı. Bu örnekte kullanılan düzen "JSteg Shell Version 2.0" internet adresinde mevcuttur [80]. 187 baytlık kısa bir mesaj tekrar beyaz. JPEG

görüntüsüne eklenilmiştir. Bu testte görüntü üzerinde fark edilebilir hiçbir nesne yoktur, lakin bu görüntünün değiştiğine dair diğer ipuçları vardır. Gömülü görüntü ile dosya boyutu 244 bayt arttı ve görüntülerin incelenmesi onun iki renkten (orijinali bir rengi belirtmektedir) oluştuğunu göstermektedir. Buna ilaveten, histogram içinde bir değişiklik vardır.

Şekil 2.10. Metin gömülü olan beyaz görüntünün histogramı. (Histogramın sağında iki rengi belirten iki adet zirve noktası vardır).

Sonuç olarak, basit denemelerin hazırlanmasını tamamlamak için, GIFshuffle denen internet adresinde mevcut bir uygulama [81] ".GIF" görüntüsü üzerinde kullanılmaktadır. Orijinal ikil eşlemli beyaz görüntü ".GIF" görüntüsüne dönüştürüldü. 187 baytlık gömülü kısa mesajdan sonra, dosya boyutu (2,402 bayt) değişmedi. Renklerin sayılarındaki kontrol, her iki görüntü içlerinde bir rengin olduğunu göstermektedir. Görüntünün değiştiğini belirten tek gösterge daha önce bahsedilen onaltılık sistem (hexadecimal) dump aracının kullanmasıydı. Bu testler herhangi yeni bir zemini bozmamışken, O çok iyi steganografi uygulamalarının olduğunu betimlemeye hizmet etti ve bazıları en az güvenirlikteydi. Ticari yazılım şartıyla, " Gözükmeyen Sırlar 3" diye bilinen bir uygulama internet adresinde mevcuttur [82]. Bu uygulama JPEG, PNG, BMP, HTML, ve wav dosyaları içindeki gizli dosyalar için bir düzenek önermektedir. Bu uygulamalar dosyaları tamamıyla ortadan kaldırmak için "güçlü şifreleme" ve "kağıt öğütücü" içermektedir.

2.3.4. Diğer steganografik yaklaşımlar

Diğer ortamlarda verileri saklamanın olası yolları nelerdir? Bir kişi farklı yaklaşımları analiz etmek için zaman ayırırsa, olası uygulamalara sınırsız bir yaklaşım gözlemleyebilir. Rowland tarafından sunulan rapora göre, TCP protokol takımı zararsız paketlerin gizli kanalları kullanarak veri geçişine izin veren zayıflıklara sahip olduğunu belirtmekte ve teorik ile uygulama eksersizlerindeki zayıflıkları örneklemeye açıklamaktadır [48].

Sui ve Luo (2004), yardımlı metin (hypertext) içerisine biçimleme (Markup) etiketlerinin pozisyonlarını değiştirerek veri sakladıklarını rapor etmişlerdir [102]. HTML kodlu web sayfalarında kullanılan etiketler (tag) aracılığıyla sayfanın görüntüsü bir biçimleme programlama diliyle şekillendirilir. HTML'de etiketlerin açılış ve kapanış bölümlerinin sırası önemsizdir. Diğer bir ifadeyle bunların sırası oluşan görüntüyü etkilemez. Örneğin:

<td width="25%" > <u> <b> <i> GAZİ<|i> <|b> <|u> <|td>

satır ile

<td width="25%" > <u> <b> <i> GAZİ<|b> <|i> <|u> <|td>

satırı aynı görüntüyü oluşturacaktır.

XML, HTML gibi bir biçimleme (markup) dilidir. Ancak HTML'den farklı olarak veri göstermeye değil tanımlamaya odaklı olarak tasarlanmıştır [103]. XML içerisine, aynı işlevi yerine getiren kodlanmasıyla veri saklanabilir. Örneğin imge etiketin için <img> </img> yazımı ile <img/> yazımı aynı işlevi görmektedir. Bu durumda <img> </img> 1, <img/> ise 0 olarak kodlanacak olursa aşağıdaki XML dosyasına ait bölümde 00101001 bitlerden oluşacak bir karakterlik bir veri saklanmış olacaktır:

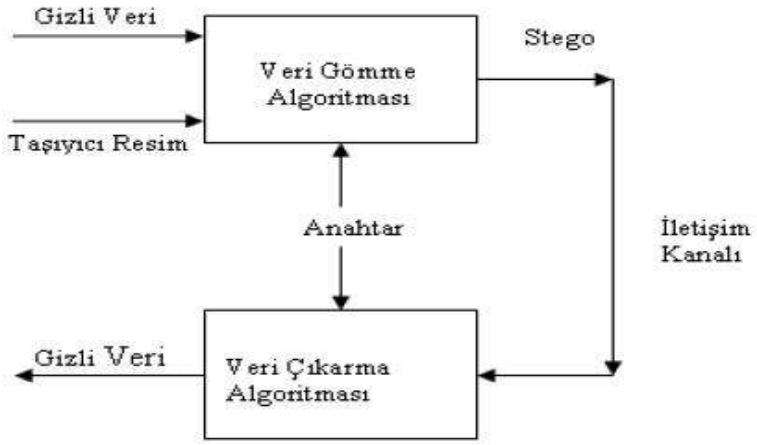
**
**
<img src=”resim3.jpg”> </img>
**
<img src=”resim5.jpg”> </img>
**
**
<img src=”resim8.jpg”> </img>

Çalıştırılabilir EXE dosyaları içerisine veri saklama aynı işlevi gören komutların varlığından faydalanarak, çalıştırılabilir dosyalar içerisine de veri saklanabilmektedir. Hydan isimli program i386 komut setindeki tekrarlardan faydalanarak 1/10 kapasiteyle bir mesajı bir uygulama içerisine saklayabilmektedir [104].

ISDN iletişimi içerisinde bilgi saklama, olası bilgi saklama alanlarından biridir. ISDN video konferansı boyunca GSM kanallarını kullanma potansiyeli tartışılmaktadır. Buna ek olarak raporda, ISDN hatlarında telefon görüşmelerinde LSB’nin içerisinde verilerin saklandığı stegonagrafi programı olarak adlandırılan DigiStilz anlatılmaktadır. Bilgiler kesinlikle her tarafa yayılmış, geniş bir şekilde değildir. Birisi mesajın nereye gizlendiğine göz atsa, mesajın digital veya analog olup olmayacağının sınırsız sayıda olma ihtimalini fark edecektir.

2.4. Steganografik Yaklaşımlar (Resim Tabanlı)

Steganografi, gizli verinin masum içeriğe sahip ve taşıyıcı olarak adlandırılan bir başka veri içersinde saklanması olarak tanımlanabilir. Bu saklama işlemi için kullanılan algoritma gizli değildir ancak güvenlik algoritma için kullanılan anahtar ile sağlanmaktadır. İçersinde gizli veri bulunduran taşıyıcıya “stego” adı verilmektedir. Stego herhangi bir yolla alıcıya ulaştırılır. Alıcı veri saklama algoritmasının tersi olan bir işlemle gizli veriyi, anahtarı da kullanarak ortaya çıkarır [49].



Şekil 2.11. Steganografi Sistemi

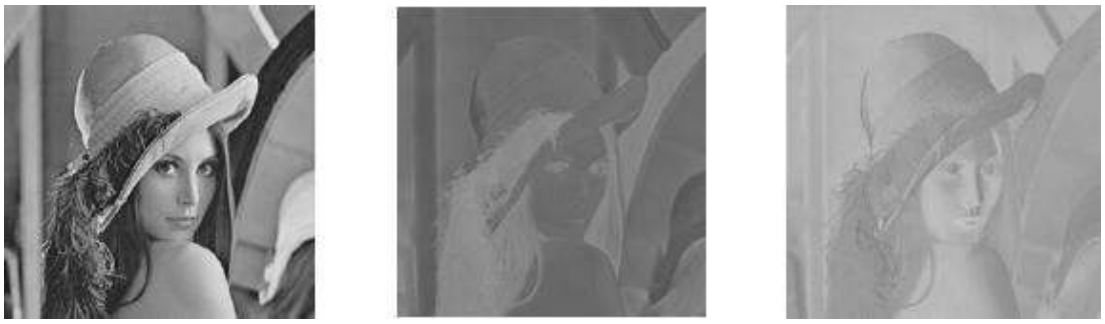
Şekil 2.11'de verilen resim içerisine veri saklama yöntemleri aşağıda sırasıyla açıklanacaktır.

2.4.1. Yer değiştirme yöntemine dayanan steganografik yaklaşımlar

Bu grupta yer alan yöntemler temel olarak resim dosyasında renkleri temsil eden değerler üzerinde çalışmaktadır. Sayısal ortamda birçok resim dosya türü bulunur. Bu dosyalar temelde resimleri sıkıştırıp sıkıştırmadıklarına göre iki gruba ayrılabilirler. Ayrıca, her resim dosyası türü renk değerlerini farklı şekilde saklı tutabilirler. En yaygın türler parlaklık bilgisinin ön planda tutulduğu YCbCr ve HSV(HSL), renklerin ön planda tutulduğu RGB'dir. Yine bu türlerin her bir renk bileşenini nasıl tuttuğu ayırt edici olabilir. Örneğin üç renk değerinin de bir sekizlide tutulduğu RGB resim dosyaları ile her bir renk değerini ayrı üç adet sekizlide saklayan RGB resim dosyaları çeşitli yönden farklıdır. En önemli fark, temsil edilebilecek renk sayısıdır. Belirtilen ikinci temsil yönteminde çok daha fazla renk değerini barındıran resim dosyalarını saklamak mümkündür. Ancak, resim dosyasının bu bilgileri saklamak için çok daha fazla yere ihtiyaç duyacağı da açıktır [50].

Yer değiştirmeye dayalı steganografik yöntemlerde çalışma ilkesi oldukça basittir. Renk değerinin tek bir sekizlide tutulduğunu varsayalım. Bu durumda 256 adet farklı

renk değerini resim dosyasında temsil edebiliriz. Adı geçen yöntemde renk değerinin en düşük anlamlı biti ile gizli verinin bitleri değiştirilir. Sonuçta ortaya çıkacak resim dosyasındaki renk değerleri ya olduğu gibi kalır ya bir artar ya da bir azalır. Ancak, her üç durumda da söz konusu olan, bu durumun insan gözü tarafından algılanamamasıdır [51].



(a) (b) (c)

Şekil 2.12 Lena'nın görüntüsü
a) Y bileşeni
b) Cb bileşeni
c) Cr bileşeni

(a) (b) (c)

Şekil 2.13 Lena'nın diğer görüntüsü
a) H bileşeni (HSV)
b) S bileşeni (HSV)
c) V bileşeni (HSV)

Yer değiştirmeye dayalı yöntemlerde, yer değişimi için kullanılacak bitlerin sayısı bir bitten daha fazla olabilir. Ancak, karar vermede, kullanılan resmin özelliği belirleyicidir. Resim dosyasında düz alanlar fazlaysa yapılan yer değiştirme işlemi çıplak gözle bile fark edilebilir. Öte yandan, steganografide unutulmaması gereken

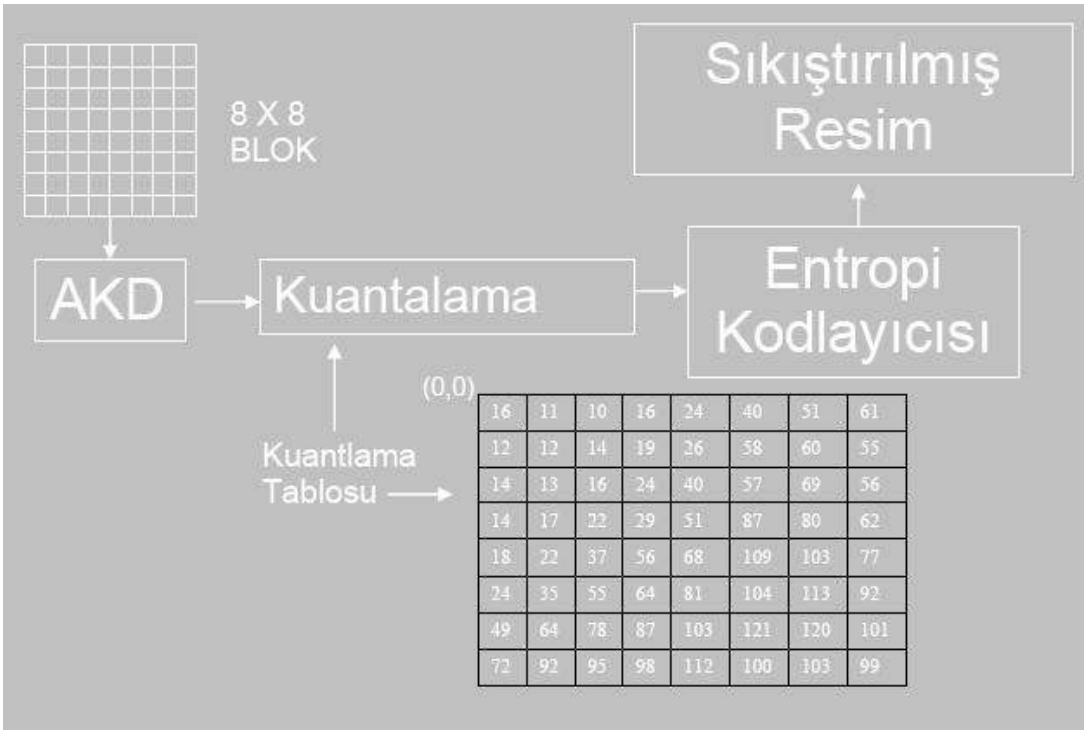
en önemli ilke, taşıyıcı dosyanın bir kere kullanılması ve sonra yok edilmesidir. Yer değiştirmeye dayanan yöntemler temelde aynı ilke ile çalışmakla beraber birden fazla şekilde gerçeklenebilirler. Her bir renk değerinin değiştirildiği gerçeklemelerle beraber renk değerlerinin gruplanarak, eşlik değer olarak gizli veri bitinin kullanıldığı ve bu grubun eşlik bitine uyum sağlaması için gruptaki tek bir renk değerinin değiştirildiği yöntemler örnek olarak sayılabilir. Aralarında ki tek fark saklanacak veri miktarıdır. Yer değiştirme yöntemine dayanan gerçeklemeler oldukça kolay bir şekilde programlanabilir. Sunmuş oldukları yüksek veri saklama miktarı ile gizli iletişim için oldukça ilgi çekicidirler. Ancak, bu yöntemlerin karşılarında iki önemli engel vardır: bunlardan ilki resim dosyaların Internet trafiğinde yaratmış oldukları yükü azaltmak için sıkıştırılmaları ve yine bu yöntemlerin sıkıştırma başta olmak üzere en hafif resim işleme yöntemlerine karşı oldukça zayıf olmaları. Gizli veri bitlerinin taşıyıcı resme gürültü olarak eklendiğini hatırlayacak olursak, parlaklığın bile değiştirilmesi bu bilgilerin tamamen yok olmasına neden olacağı açıktır.

2.4.2. İşaret işleme yöntemine dayanan steganografik yaklaşımlar

Günümüz bilgi dünyasında Internet en önemli bilgi kaynağı olarak algılanmaktadır. Bu inanılmaz büyüklükteki veritabanının ihtiyaçlara en uygun şekilde cevap verebilmesi için en önemli koşullardan biri sunucu ve istemci arasındaki iletişimin mümkün olduğu kadar kısa bir sürede, son kullanıcıyı memnun ederek sonlanmasıdır. Bu nedenle Internet trafiğinin düzenlenmesi konusunda oldukça fazla çalışma yapılmaktadır. Ancak, bilgiyi paylaşanlar en başta yaptıkları iyileştirici uygulama, Internet'e sunulan grafik dosyaların sıkıştırılmasıdır. Kaliteden biraz ödün vererek grafik dosyalarının büyüklüğü ile ciddi miktarda oynamak mümkündür. Internet kullanıcıları var olan resim sıkıştırma algoritmaları ile yüksek oranda verim alabilmektedir. Bu sıkıştırma algoritmaları içinde JPEG, GIF, ve PNG önemli yer tutar [52].

Tüm sıkıştırma algoritmaları insan gözünün filtreleme özelliğini kullanır. İnsan gözü belli bir frekanstan sonrasında ki renk değişimlerini algılayamaz. Dolayısıyla, asıl

resimdeki renk değerleri frekans düzlemine taşınabilir. Frekans düzleminde kullandığımız dönüşüme göre çeşitli katsayılar karşımıza çıkacaktır. Bu katsayılar kullanılarak ters işlem sonrası tekrar asıl resmi elde etmek mümkündür. Ancak, asıl resmi elde etmek için sonsuz sayıda frekans bileşeninden yararlanmak gereksizdir. Belirli bir kaybı göze alarak yüksek frekans katsayılarından vazgeçilebilir. Vazgeçilen frekans sayısı tekrar elde edilen resmin kalitesini ortaya koyacaktır. Yapılan uygulamalarda resim kalitesinden küçük miktarlarda kalite kayıplarına karşın resim boyutunda 1:100 oranında küçültme mümkün olmuştur. Sıkıştırma algoritmaları içerinde yaygın olarak kullanılan JPEG sıkıştırma algoritması da benzer bir ilkeyle çalışmaktadır.



Şekil 2.14 JPEG Sıkıştırma Algoritması [97]

Şekil 2.14'de verilen şekilde de özetlendiği gibi JPEG sıkıştırması ilk önce resmi 8x8 renk değerinden oluşan parçalara ayırır. Her biri üzerinde frekans düzlemine geçmek üzere ayrık kosinüs dönüşümü (AKD) uygular. Bu dönüşümün tercih edilmesinin sebebi katsayıların karmaşık değil reel sayılar olmasıdır. Elde edilen sayılar seçilen kalite oranına göre seçilen bir çizelge kullanılarak kuantalanır. Kuantalam çizelgesi

kaliteye göre yüksek frekans katsayılarından kaç tanesinin yok edileceğini belirler. Kuantalama işlemi sonucunda ayrık kosinüs dönüşümü sonrası elde edilen 64 adet frekans katsayılarından bir kısmı sıfır değerini alacaktır. Bu katsayıların Huffman kodlaması ile sıkıştırılması ciddi oranda yer kazanımı sağlar [53].

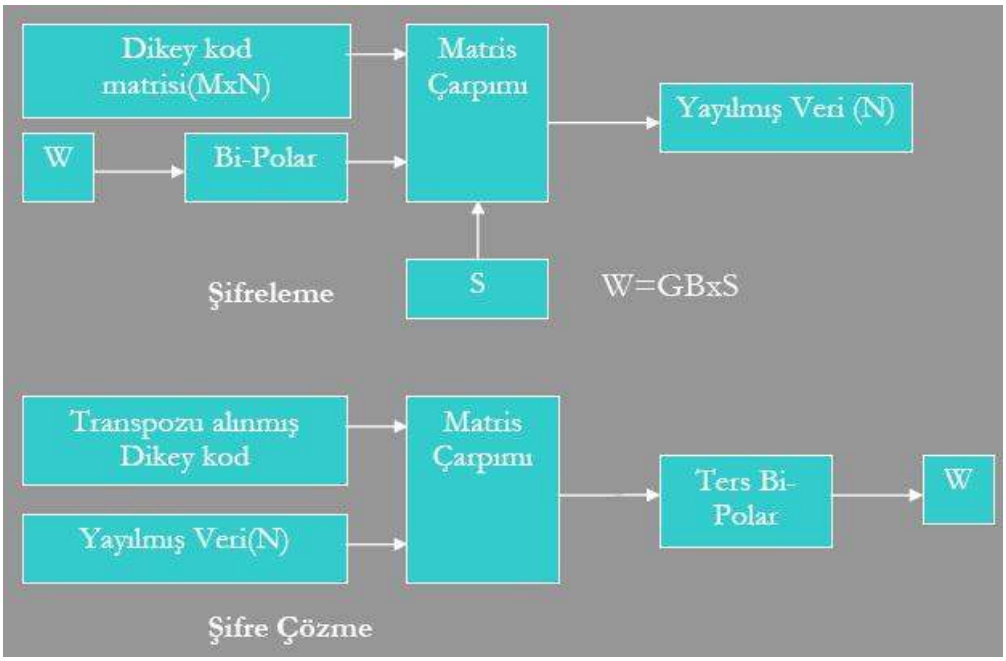
Resim tekrar elde edilmek istenildiğinde sıkıştırma işleminin tersi uygulanacaktır. İlk önce Huffman kodu çözülür ve içerisinde sıfır katsayılarının da olduğu 64 adet katsayıdan oluşan blok elde edilir. Bu blok yine aynı kuantalama çizelgesi ile çarpılır. Birçok katsayı asıl değerine yakın değerlere ulaşacaktır ancak sıfır ile çarpılanlar sıfır kalacaktır. İşte bu, ters kosinüs dönüşümü sonrası elde edilen 8x8 renk değerlerindeki kaybı belirleyecektir.

Sıkıştırma işleminde kullanılan yöntem, steganografik bir algoritma oluştururken bize yol göstermektedir. Her şeyden önce 8x8 renk bloklarının dikkate alınması gerekir. Öte yandan sıkıştırma işlemi kayıplı olacaktır. Bu işlemlere dayanıklı bir algoritma ayrık kosinüs dönüşümünü dikkate almalıdır. Önerilen birçok yöntemde, algoritmalar ayrık kosinüs dönüşümünü kendileri yapmaktadırlar. Daha sonra yok edilecek frekans katsayılarını dikkate alarak her bir blokta 1 bit saklarlar. Burada önemli olan, düşük frekans değişimlerinin resmi bozacağının kesin olmasıdır. Öte yandan, yüksek frekans katsayıları da muhtemelen yok edileceklerdir. Dolayısıyla resmi değiştirmeyecek kadar yüksek, yok edilmeyecek kadar düşük frekans katsayılarının bulunması gerekir. Ayrık kosinüs dönüşümü kullanarak gizli iletişimi sağlayan yöntemlerde ortak sorun katsayıların belirlenmesidir. Ancak, katsayılar resimden resime fark göstereceklerdir. Resimde bozulma olup olmayacağı malesef ancak işlem sonrası anlaşılabilir. Ayrıca, her 64 adetlik bloklara 1 bit gömülebilmesi, saklanabilecek veri miktarını önemli oranda düşürmektedir.

2.4.3. Spektrum yayılması yöntemine dayanan steganografik yaklaşımlar

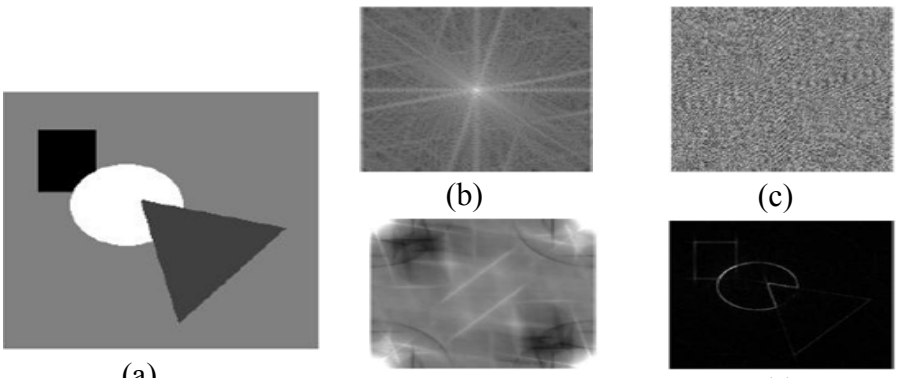
Son yıllarda yapılan çalışmalarda en dikkat çekici olanları spektrum yayılmasına dayanan yöntemlerdir. Spektrum yayılması aslında askeri iletişimde çok eskiden beri kullanılan bir yöntemdir. Bu yöntemde gönderilmek istenen mesaj ihtiyaç duyduğu

frekans bandından çok daha fazlasına dağıtılır. Üçüncü bir kişi araya girip bir ya da birden fazla frekans bandında bozulmalara neden olsa bile, alıcı geri kalan frekans bantlarındaki bilgiler ile asıl mesajı elde edebilmektedir. Aynı yöntem steganografi için de uygulanabilir. Gizli mesaj birden fazla bantta yayılarak resme gürültü olarak eklenebilir [54].



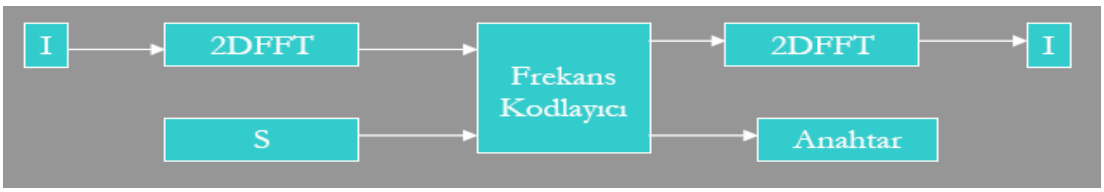
Şekil 2.15. Spektrum Yayılması Algoritması [21]

Spektrumu yayılan bilgilerin resmin renk değerlerine nasıl gömüleceği uygulamadan uygulamaya değişmektedir. Bunlardan ilkinde resmin Fourier dönüşümünü alıp genlik kısmına gürültü olarak eklenir. Önerilen ikinci bir yöntem ise bu bilgilerin resme doğrudan gürültü olarak eklenmesidir. Bu yöntemde en büyük sorun karşı tarafta değiştirilmemiş taşıyıcının tahmininin yapılması gerekmektedir. Bunun için görüntü işleme teknikleri kullanılmaktadır. Fourier dönüşümü kullanılarak verinin resme gömülmesinde dikkat edilmesi gereken nokta genlik bilgileri değiştirilirken faz bilgisinin aynen korunmasıdır. Dönüşümdeki faz bilgilerinin kısmen de olsa değişmesi durumunda ters Fourier işlemi sonucunda elde edilecek resim, taşıyıcı resim olmayacaktır. Faz ve genlik bilgilerinin önemini ortaya koyan Şekil 2.16'deki şekiller ilgi çekici olacaktır.



Şekil 2.16. Renk bilgilerden asıl resmi elde etme
a) Asıl resim
b) Genlik spektrumu
c) Faz spektrumu
d) Elde edilen resim (faz bilgisi yok)
e) Elde edilen resim (genlik bilgisi yok)

Şekil 2.16'de verilen resimlerde de görülebileceği gibi genlik bilgilerinden hareketle asıl resim elde edilememektedir. Oysa yalnızca faz bilgisi dahi resim hakkında önemli miktarda bilgi vermektedir. Yayılan gizli veri Şekil 2.17'de verilen diyagramda belirtilen yöntemle resim eklenmektedir.



Şekil 2.17. Gizli veriyi ekleme algoritması [21]

Bu yöntem ile gömülen bilgi karşı tarafta kolayca elde edilmektedir. Ancak, yolda gerçekleşecek birçok müdahaleye diğer belirtilen yöntemlere göre çok daha fazla dayanıklıdır. Resmin sıkıştırılması, bir bölümünün tamamen bozulması ve parlaklık değerleri ile oynanması durumunda bile alıcı gizli mesajı başarılı bir şekilde alabilmektedir [55].

Spektrum yayılmasına dayanan yöntemlerde kapasite kullanımı kullanılan dikey koda bağlıdır. Bu durumda örnek diyagramdan da görülebileceği gibi gizli verinin miktarının önemi yok gibi görünmektedir. Ancak, verinin sağlıklı çıkarılması için

dikey kod uzunluğunun gizli veriden çok büyük olması gerekmektedir. Bu durum gönderilebilecek veri miktarını sınırlamakla beraber yapılan çalışmalarda bu oranın ne olması gerektiği konusunda bir bilgi yoktur.

2.4.4. İstatistiksel yöntemlere dayanan steganografik yaklaşımlar

Önerilen çalışmalardan bazıları resmin bazı istatistiksel bilgilerinin değiştirilmesi ile alıcıya gizli bir mesaj verilebileceğini belirtmektedirler. Burada bir hipotez fonksiyonu belirlenir ve bu hipotez fonksiyonuna parametre olarak resim veya resmin bir kısmı gönderilir. Bu fonksiyonun geri döndürdüğü değer fonksiyonda incelenen istatistiksel özelliğin değişip değişmediğini belirtmektedir.
Bu yöntemler çeşitli resim işlemlerine dayanıklı olduklarını iddia etmektedirler. Yine bu yöntemlerde karşılaşılan sorun ise uygun bir hipotez fonksiyonunun bulunmasıdır. Ancak, en büyük sorun gönderilebilecek gizli bilgi miktarıdır. Önerilen yöntemlerde resim dosyası başına 1 adet gizli veri biti gönderilebilmektedir.

2.4.5. Diğer steganografik yaklaşımlar

Üzerinde çalışılan bir diğer yöntem ise taşıyıcı resmin, gizli veriden hareketle oluşturulma çabasıdır. Böyle bir yöntemde taşıyıcı resim hiçbir şekilde değiştirilmediği için üçüncü bir kişinin fark etmezi mümkün değildir. Ancak yine bu yöntemde en büyük sorun bu yolla üretilen resimlerin insan gözünü bile aldatamamasıdır. Düz metin steganografisi bir diğer çalışma alanıdır. Dilin yapısına uygun olarak cümle elemanlarına 0 veya 1 değerleri atanır ve gizli veri bitlerine göre seçilen kelimeler ile cümleler oluşturulur. Ancak, burada en büyük sorun sürekli bir akış sağlanmasıdır. Üretilen cümlelerin anlamlı ve birbirleriyle ilgili olması ciddi bir sorundur [56].

Resim dosyaları üzerinde gerçekleştirilen steganografik uygulamaların benzerleri ses ve görüntü dosyaları üzerinde de gerçekleştirilmektedir. Ancak, burada en büyük sorun bu tür dosyaların ciddi miktarda sıkıştırılmış olması veya görüntü dosyalarında olduğu gibi çerçevelerin fark çizelgelerini yansıtmasıdır.

2.5. Ses Dosyalarında Veri Gizleme Yöntemleri

Ses dosyalarında veri gizleme yöntemleri aşağıda alt başlıklarda açıklanmıştır [65].

2.5.1. Düşük bit kodlaması (Low-Bit Encoding)

Görüntü steganografide kullanılan LSB ekleme yöntemiyle aynı şekilde gerçekleştirilir. Ses dosyasındaki verinin her baytının son bitine gizlenecek bilginin bir biti yazılır. Sonuçta oluşan değişiklik ses dosyasında gürültüye neden olmaktadır. Ayrıca dayanıksız bir yapısı vardır. Tekrar örnekleme veya kanalda oluşabilecek gürültü ile mesaj zarar görebilir veya yok edilebilir.

2.5.2. Aşama kodlaması (Phase Encoding)

Aşama kodlaması yöntemi de resim dosyalarında uygulanan JPEG algoritması benzeri bir yapı taşımaktadır. Gömme işleminde ses dosyası küçük segmentlere bölünür ve her segmente ait aşama (faz) gizlenecek veriye ait aşama referansı ile değiştirilir. Aşama kodlaması prosedürü aşağıdaki gibidir [65]:

- Ses verisi N adet kısa segmente bölünür.
- Her segmente Discrete Fourier Transform (DFT) uygulanarak aşama ve büyüklük (magnitude) matrisleri yaratılır.
- Komşu segmentler arasındaki aşama farklılıkları hesaplanır.
- Her segment için yeni bir aşama değeri bilgi gizlenerek oluşturulur.
- Yeni aşama matrisleri ile büyüklük matrisleri birleştirilerek yeni segmentler elde edilir.
- Yeni segmentler birleştirilerek kodlanmış çıkış elde edilir.

2.5.3. Taft yayılması (Spread Spectrum)

Gizleme işlemini ses sinyalinin kullandığı frekans taftı üzerinde yapmaktadır. Güçlü bir yapısı olamamakla birlikte seste gürültü meydana getirmektedir.

2.5.4. Yankı veri gizlemesi (Echo Data Hiding)

Bilginin gizlenmesi taşıyıcı ses sinyali üzerine bir yankı eklenmesi ile sağlanmaktadır. Bilgi yankının gecikme miktarı, zayıflama oranı veya büyüklüğü gibi değerler kullanılarak gizlenir. İki farklı gecikme değeri kullanılarak insan kulağının algılamayacağı düzeyde 0 veya 1'in kodlanması mümkündür. Her bitin kodlanması için sinyal segmentlere bölünür. Yankı veri gizlemesi yöntemi herhangi bir gürültüye neden olmamakta veya kayıplı bir kodlama kullanmamaktadır.

2.6. Gelecekte Steganografi

Bir bilgi güvenliği uzmanı olan Richard E. Smith'e göre, hiç kimsenin sizden steganografiyi kullanmanızı beklemediği müddetçe çalışmasından dolayı, steganografi için birçok pratik kullanım görülmemektedir.

Devam eden çalışmalar, steganaliz yöntemi ile steganografik görüntülerin sağlamlaştırılması konusu üzerinde yoğunlaşmaktadır. Niels Provos'un "Defending Against Statistical Steganalysis" konulu makalesinde belirttiği gibi bundan sonra geliştirilecek olan yeni metotlar, bir kişiye mesajın en güvenilir biçimde saklanabileceği dosyayı seçme şansı verecek ve bu dosya standart istatistiksel analizlere karşı dirençli olacaktır.

Steganografik tekniklerin belirli kullanımları vardır. Bazıları yasal, bazıları kısmen yasal bazıları ise yasal olmayan kullanımlardır. Mülkiyetin, hakikiliğin ve zihinsel fikirlerin korunması için kuvvetli bir iştir [57].

Sayısal ortam damgalama (digital medya watermarking) yöntemi, öncelikli olarak sahibini kanıtlamak için sayısal damga kullanarak sürekli bir şekilde gelişim göstermektedir Bireyler veya organizasyonlar, kişisel, özel ve hassas bilgilerini steganografik taşıyıcılar üzerinde bulundurmaya karar vereceklerdir. Herkesin kabul edebileceği gibi, bunu başarmak için genellikle pek çok yöntem mevcuttur. Bazıları bu uygulamaları bir kapı kilidinin kullanımına benzetebilir. Kilit, dürüst insanları

koruyacaktır fakat bir kişi eğer içeri girmeye niyetliyse pencereyi kırarak içeri basitçe girebilir.

Steganografideki gelişmelerle birlikte bilgisayar ve iletişim ortamının oldukça güvenli bir depolama ve iletim yöntemleri sunması mümkün olacaktır.

Steganografinin diğer kullanım alanları pornografiden virüslere kadar birçok kötü uygulamalarda söz konusudur. Çocuk pornografisinin internet üzerinde masum görüntü veya ses dosyalarının içerisine gizlenmiş olduğu iddiaları mevcuttur [106]. Bu iddia bütünüyle doğru olsa bile, bu iddianın doğruluğunu internette araştırabilecek yöntemler başarısızla sonuçlanmıştır. High Technology dergisi tarafından yayımlanan bir rapora göre, aşağıdaki bilgisayar suçları çeşitleri en yaygın karşılaşılan ilk dokuz suçları göstermektedir [107];

• Suçla ilgili (kriminal) haberleşmeler
• Dolandırıcılık
• Hacking (Kırma)
• Elektronik ödemeler
• Kumar ve pornografi
• Usandırma ve rahatsız etme
• Zihinsel mülkiyete yapılan saldırılar
• Virüsler
• Çocuk istismarı

Yukarıdaki liste incelendiğinde, birçok alanda özellikle kriminal haberleşmede steganografi yöntemleri yaygın olarak kullanılmaktadır. Eğer bilgisayar tabanlı uygulamaların dışında steganografik yöntemler araştırılırsa,kullanım potansiyeli oldukça fazla çıkacaktır. Bilgisayar güvenliği açısından, bazı alanların varlığından farkında olunması gerekmektedir. Potansiyel içeri sızmalara sahip bir alan " Ağ seviyesinde güvenliği atlatmak için steganografi kullanan bir protokol" olarak tanımlanabilir. Peacefire.org (18 yaşın altındaki kullanıcıları hedefleyen sansüre karşı bir organizasyon)'un koordinatörü olan Bennet Haselton, bu protokolü tespit

edilemeyen sansürler olarak tanımlamaktadır.

Son olarak, bilgisayar savaşları ele alınmalıdır. Jordon T. Cochran yüksek lisans tezinde, USAF kuruluşunun steganografik virüs saldırılarını araştırdığını belirtmektedir [108]. Yaptığı çalışmada Cochran şöyle demektedir; "Sonuçlar göstermektedir ki steganografik araçlar sadece saldırı silahlarına yardım etmemektedir. Fakat diğer uygulamalarla birleşince bu araçlar, minimal kullanıcı müdahalesi ile birlikte otomatik olarak gizli bilgiyi ortaya çıkartmak için kullanılabilinir" [108].

Diğer bir tezde Dale A. Lathrop, ayrıca steganografik teknikleri kullanarak virüs saldırılarının olma ihtimalini araştırmaktadır [84]. Tezinde şunu ortaya koymuştur. Bu çalışmanın sonuçları da göstermiştir ki "Steganografik yöntemlerin gömülü olduğu virüs ve diğer payloadlarla birlikte Fotoğrafik bir görüntü içeren HTML tabanlı elektronik mail mesajlarını takip eden ayrı bir motorun kullanılması eğer uygun ortam değişkenleri yerleştirilmeden uygulanmışsa virüs saldırılarına her zaman açıktır." Daha sonra yapılan çalışmalarda ortaya koymuştur ki "Virüs saldırılarını başlatmak için hala daha insan müdahalesine ihtiyaç vardır" [84].

Elektronik suçlarla ilgili detaylı bir araştırma "Electronic Crime Scene Investigation, A Guide for First Responders" başlığı altında Temmuz 2001'de yayımlanmıştır ve bu raporda elektronik suç olaylarını tespit etmek ve korunmak için temel ve sağlam bazı tavsiyelerde bulunulmuş ve steganografiye önem verilmiştir [88].

2.7. Steganografik Uygulamalar Hakkında Genel Değerlendirmeler

Steganografi uygulamalarında kullanılacak dosya türünde bir sınırlama olmamakta, dosyanın yapısal özellikleri uygun olduğu müddetçe her türlü sayısal dosya veri saklamada taşıyıcı dosya olarak kullanılabilmektedir. Örneğin MS Word dosyalarında metin rengiyle sayfa renginin aynı yapılmasıyla bir mesaj basitçe gizlenebilir ve bu durum steganografik amaçlı kullanılabilir. Steganografi uygulamalarında kullanılan taşıyıcı dosya türlerindeki çeşitlilik her geçen gün

artmaktadır. Saklama işlemi sonucu oluşan yeni dosyanın şeffaflığının korunması ve saklanan verinin doğru bir şekilde tekrar geri getirilmesi sağlandıktan sonra saklama kapasitesi ve dosyanın iletişimdeki popülerliğini de göz önüne alarak, yapısı uygun her türlü dosya steganografi amaçlı kullanılabilir. Çizelge 2.2'de steganografik uygulamalarda kullanılan bazı dosya türleri yer almaktadır. Çok farklı sayısal dosya türleri içerisine veri saklama çalışmalarının gelecekte daha da artacağı değerlendirilmektedir [23].

Çizelge 2.2. Çeşitli taşıyıcı dosya türleri

	Dosya Uzantısı	**Açıklama**
1	BMP	Bitmap
2	JPEG	Joint Photographic Experts Group
3	PNM	Portable Any Map
4	GIF	Graphics Interchange Format
5	WAV	Waveform sound
6	TXT	Metin dosyası
7	HTML	Hyper Text Markup Language
8	XML	eXtensible Markup Language
9	PCX	PC Paintbrush Graphic
10	PNG	Portable Network Graphic
11	AIFF	Audio Interchange File Format
12	MP3	MPEG-1 Audio Layer III
13	PDF	Portable Document Format
14	MIDI	Musical Instrument Digital Interface
15	EXE	Executable
16	DLL	Dynamic Link Library
17	TIFF	Tagged Image Format
18	DOC	Document

2.8. Steganografik Sistemi Değerlendirme Kriterleri

Bir steganografik algoritma değerlendirilirken 3 temel özelliği dikkate alınır. Bunlar;

− Taşıyıcıdaki değişim
− Kapasite
− Dayanıklılık'tır.

Resim üzerinde gerçekleştirilen değişiklikler insan gözü tarafından algılanmamalıdır. Aksi durum, birinci ilkenin ihlali ve gizli iletişimin ortaya çıkmasıyla sonuçlanacaktır. Öte yandan iletişim her ne kadar gizli de olsa üçüncü bir kişinin içerisinde gizli veri olan dosya üzerinde bazı işlemler yapma olasılığı vardır. Steganografik yöntemin bazı sınırlar içerisinde de olsa bu tür saldırılara karşı belli bir dayanıklılık sergilemesi gerekir. Steganografik algoritmalarda sağlanmak istenen, yüksek miktarda gizli verinin saklanabilmesidir. Malesef, var olan algoritmaların tüm bu koşulları tam olarak sağlayamadığı bir gerçektir. Daha da kötüsü, bu koşullar arasında ciddi ikilemler vardır.

Taşıyıcı içerisine yüksek miktarda veri gömebilmek istenilen bir özelliktir. Ancak, veri miktarı arttıkça resim üzerindeki değişim insan gözüyle algılanabilecek sınırlara ulaşabilmektedir. Bu da kapasite ve değişim arasında bir ikilem yaratmaktadır. Yine benzer şekilde gömülen gizli verinin yapılan saldırılar sonucunda bile alıcı tarafında başarıyla çıkartılabilmesi için düşünülen dayanıklılığı arttırıcı önlemler ve/veya yöntemler yapılması gereken işlem miktarını arttırmakta ve yine dosya üzerinde yapılan değişimi belli edebilecek seviyeye ulaşabilmektedir. Tüm bu koşullar ve ikilemler dâhilinde her türlü saldırıya göğüs gerebilecek bir algoritma henüz yayınlanmamıştır. Ancak, çeşitli düzeydeki ihtiyaçlar doğrultusunda önerilen algoritmalar başarılı olarak kullanılabilir.

2.8.1. Taşıyıcıdaki değişim

Bir steganografik algoritma değerlendirilirken taşıyıcıda (cover object) ne kadar

değişim olduğu da çok önemlidir. Taşıyıcıdaki değişimi ya da resimdeki bozulma oranının belirlenmesi için çeşitli ölçme yöntemleri vardır. Bunlar arasında en bilinenleri; MSE, RMSE, PSNR' dır [95].

MSE (Mean Squared Error) hataların kareleri toplamının ortalamasıdır. MSE genellikle σ^2 olarak gösterilir. RMSE (Root Mean Squared Error) ise MSE'nin karekköüdür [33].

$$\sigma^2 = \frac{1}{\mathrm{N}} \sum_{n=1}^{N} (x_n + \mathrm{y_n})^2 \tag{2.1}$$

Bazen MSE yerine, hatanın büyüklüğünün orijinal piksel değerinin en büyüğü (peak-tepe) ile olan ilişkisi ile ilgilenilir. Bu gibi durumlarda PSNR (Peak Signal-to Noise Ratio) yöntemi kullanılmaktadır [33].

$$PSNR(dB) = 10 \log_{10} \frac{X_{peak}^2}{\sigma_d^2} \tag{2.2}$$

JPEG ve GIF formatındaki resim dosyaları özelliklerinden dolayı taşıyıcıdaki değişime daha duyarlıdır. Taşıyıcıdaki değişim açısından BMP dosyalar daha iyi sonuçlar vermektedir.

2.8.2. Kapasite

Sıralı LSB yönteminde kapasite resmin boyutuyla ilgilidir. Bu yüzden aynı büyüklükte resimler için bu yöntemlerin saklayabilecekleri bilgi miktarları eşittir.

Kapasite açısından da BMP ve GIF formatındaki dosyalar daha iyi sonuçlar vermektedir. JPEG formatındaki dosyalarda 8x8 piksellik bloklara sadece 1 Byte saklanabilmektedir. Bu yüzden saklanabilecek veri miktarı oldukça azdır.

2.8.3. Dayanıklılık

Bir steganografik sistemin dayanıklılığını ölçmek için steganaliz yöntemleri kullanılmaktadır. Steganaliz, bir örtü verisi (cover data) içerisinde herhangi bir bilgi olup olmadığını bulmayı ve eğer var ise bu bilgiyi elde etmek amacıyla steganografik algoritma kullanılan sisteme karşı yapılan saldırı yöntemleridir.

Genelde saldırı yapan kişinin (steganalist) kullanılan steganografik sistemi bildiği varsayılır (*Kerchoffs'un prensibi*). Eğer steganalist kullanılan sistemi bilmiyorsa, bu onun işini zorlaştıracaktır [34]. Steganalistin bir steganografik sisteme saldırabilmesi için sahip olması gereken veriler vardır. Bu sahip olduğu verilere göre saldırı modellerinden birini seçebilir.

Bu saldırı modellerinden en yaygın olanları şunlardır [37,38,39] :

1. Sadece stego saldırısı: Analiz için sadece stego-nesnesi (Stego-object) (Görüntü dosyası) bilinmektedir.
2. Bilinen cover (örtü) saldırısı: Görüntünün mesaj gizlenmeden önceki ve sonraki hali bilinmektedir.
3. Bilinen mesaj saldırısı: Saklanan mesaj bilinmektedir.
4. Seçilmiş stego saldırısı: Steganografik algoritma ve stego-nesnesi bilinmektedir.
5. Seçilmiş mesaj saldırısı: Steganalist bu yöntemde stego-nesnesini analiz edebilmek için çeşitli mesajlar seçer, steganografik araçlar kullanır ve algoritmayı bulmaya çalışır.
6. Bilinen stego saldırısı: Örtü nesnesi, stego nesnesi ve steganografik araçlar bilinmektedir.

Her steganografik yöntem için ayrı steganaliz yöntemleri geliştirilmiştir. Bir yöntem için çok iyi sonuçlar veren bir steganaliz yöntemi bir diğeri için doğru sonuç vermemektedir. Dayanıklılık ölçütünde JPEG formatı için kullanılan steganografik yöntemler daha başarılıdır. Çünkü steganalitik saldırılara daha dayanıklıdır.

Öncelikle resmin içinde veri gizlenip gizlenmediğini anlamak için sezme (detection) saldırıları yapılır. Bu saldırı yöntemleri [92];

- $\chi 2$ Testi
- Histogram Analizi
- RS Steganalizi
- RQP Yöntemi
- Görsel Ataklar
- JPEG Steganaliz
- Evrensel tespit sistemleridir

şeklinde sınıflandırılabilir.

Resmin içinde veri olduğu anlaşılırsa, bu veriyi elde etmek amacıyla çekme (extraction) saldırısı yapılır [37].

Eğer resmin içindeki gizli veri bozulmak isteniyorsa resmin içinden bir parçayı kesip çıkarmak ya da resme başka bir veri daha gizlemek gibi saldırı yöntemleriyle de resim içindeki gizli bilgi etkisiz ve işe yaramaz hale getirilebilmektedir.

Bu konular tezim ana konusu olduğu için üçüncü bölümde daha detaylı incelenmiştir.

3. STEGANALİZDE KULLANILAN TEKNİKLER

Steganaliz konusunda birçok çalışma yapılmıştır [83,39,85,86,87,4,89, 90,91,111].

- Westfeld ve Pfitzmann (2000). resimlere mesajın saklanması sonucu değişen değer çiftlerinin istatistiksel analizine dayanan bir steganaliz metodu geliştirdiler. Bu yöntem ardışık saklama gibi durumlarda daha güvenilir sonuçlar vermiştir [4].

- Fridrich ve ark. (2000), 24 bitlik renkli resimlerde LSB yöntemiyle gerçekleştirilen veri saklamayı tespit etmek için, bir birine yakın renk değeri çiftlerinin istatistiksel analizine dayanan bir çalışmada tekil renk oranının toplam piksel sayısının %30 undan az olduğu resimlerde güvenilir sonuçlar verebilen bir yöntem sunmuşlardır. Bu yöntemin sıkıştırılmamış yüksek çözünürlüğe sahip, yani günümüzdeki sayısal fotoğraf makinelerinin çektiği resimlerde başarılı olamayacağı ve ayrıca gri seviyeli resimlere bu yöntemin uygulanamayacağını bildirmişlerdir [39].

- Provos ve Honeyman (2001), JPEG resimlerinin steganalizi üzerine kapsamlı bir çalışma yaptılar ve geliştirdikleri yazılımla internet üzerinde binlerce şüpheli resmi tespit edebilmişlerdir [111].

- Farid (2001), taşıyıcı ve stego resimlerden oluşan bir veritabanı üzerinde yaptığı analizlerden ve özellik çıkarımlardan sonra steganografi algoritmasının bilinmesine ihtiyaç duymaksızın steganaliz tahmininde bulunan evrensel tespit çalışmasını sunmuştur [87].

- Fridrich ve ark. (2001), renkli ve gri seviyeli resimlerin steganalizi üzerine RS ismini verdiği bir metot geliştirmişlerdir. Bu metotta veri saklama sırasında değişen değer çiftlerinin aynı sıra bu değişimin gerçekleştiği piksel konumlarının da istatistiksel analizi yapılmıştır. Bu yöntemin başarısı da gömme işleminin rastgele olduğu durumlarda ardışık saklamaya göre daha düşüktür [85, 86].

- Fridrich (2006), veri saklama sonucu oluşan değişim ile değişim sayısı arasındaki ters orantıyı analiz ettiği çalışmasında, herhangi bir tespit edilebilirlik profilinde, (tespit edilebilirlik profili gömme etkisinin dağılımını ifade etmektedir) sadece en küçük gömme etkisine sahip piksellerin kullanılmasının asla optimum olmadığını belirlemiştir [89].

Resimler için steganaliz yaklaşımlarından bazıları şunlardır [92]:

- Görsel tespit (Görsel ataklar)
- $\chi 2$ Testi
- Histogram analizleri
- Yüksek seviyeli istatistiksel analizler (RS analizi), (İkili istatistik yöntemi)
- Algoritmaya ve türe özel tespitler (JPEG Steganaliz)
- Evrensel tespit sistemleridir.
- RQP Yöntemi.

Görsel tespit, veri gömme sonucu oluşan resmin gözle orijinalinden farkının anlaşılması veya orijinali yoksa resmin üzerindeki normal bir resimde olması beklenmeyen bozuklukların fark edilmesiyle gerçekleşir. Ancak bu durumda da kendinden gürültülü resimlerle stego resimleri ayırt edebilmek mümkün olmayacaktır. Görsel atakların güvenilirliğinin tartışılır olduğu açıktır.

Histogram analizleri resme veri gömme sonucu değişen piksel değeri çiftlerinin istatistiksel analizlerini esas almaktadır.

Yüksek dereceli istatistiksel analizler ise histogram analizinden fazla olarak değişimin gerçekleştiği piksellerin konumlarına yönelik istatistiksel analizleri de içermektedir.

Algoritmaya veya türe özel tespitler de algoritması bilinen steganografi uygulamalarıyla oluşturulan stego resimlerdeki mesajı ortaya çıkarmaya yönelik yöntemlerle JPEG gibi resim türlerine özel steganaliz çalışmalarını içermektedir.

Evrensel tespit sistemleri ise resme ait bazı özelliklerin çıkarımıyla orijinal resimlerle stego resimler arasındaki ayırımı yapmaya yönelik yöntemleri içermektedir.

RQP Yöntemi Fridrich ve arkadaşları tarafından geliştirmiştir [39]. Bu metot LSB gizlemesi tarafından yaratılan yakın renk çiftlerini analiz etmeye yöneliktir. Öncelikle seçilen resim için yakın renk çiftlerinin tüm renk çiftlerine oranı hesaplanır. Daha sonra bu resim içerisine bir test mesajı gizlenerek oran yeniden hesaplanır. Bu iki oran arasındaki fark büyük ise resminin içinde gizlenmiş bilgi yok demektir. Bu iki oranın birbirine yakın olması resmin içinde gizlenmiş bilgi olduğunu göstermektedir.

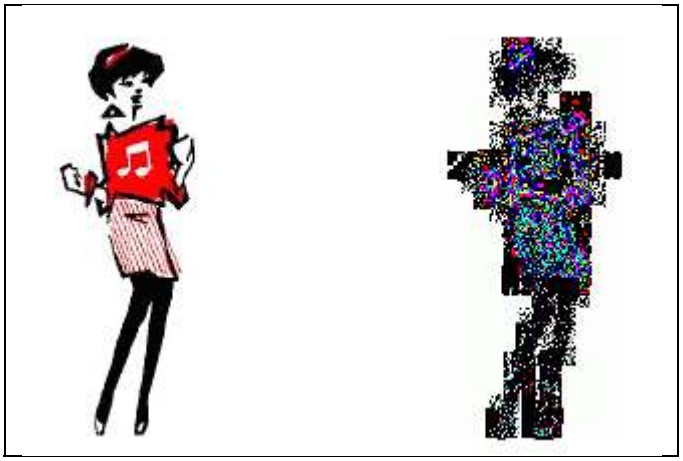
3.1. Görsel Saldırılar

Birçok steganografik program mesaj bitlerini sıralı veya bir takım sözde rastgele şekilde gömerler. Birçok programda, mesaj bitleri görüntü içeriğinden bağımsız seçilir. Eğer görüntü birleştirilmiş düzenli renk alanları 0 veya 255 miktarında doymuş renk alanları ihtiva ediyorsa, stego görüntüyü ön işlemden geçirdikten sonra şüpheli görüntüyü basit görsel incelemeyi teknikleri kullanıralak tesbit edebiliriz. Her ne kadar bu eserler kolayca görülemezse de bir bit düzlemi (örneğin LSB düzlemi) çizebilir ve bu bit düzleminin kendisini inceleyebiliriz. Bu saldırı özellikle palet indislerinde gömülü LSB palet görüntülerine yapılmaktadır. Eğer aynı zamanda mesaj sıralı olarak gömülmüşse, görüntüde steganografik mesajların varlığı için ikna edici bir delil vardır. Bununla birlikte, Pfitzmann ve Westfeld'in bildirdiği şekilde [4], gürültülü görüntüleri veya yüksek dokulu mesajları stego görüntülerden bu tekniği kullanarak ayırmak mümkün değildir. Görsel saldırılar basit olmasına rağmen, bunları otomatikleştirmek zordur ve güvenilirlikleri çokça sorgulanır.

Görsel ataklarda temel fikir, görüntünün içinde mesaj olan tüm parçalarının görüntüden çıkarılmasıdır. Bu şekilde insan gözü görüntünün içinde bir mesaj olup olmadığını ayırt edebilecektir [4]. Görsel atakların amacı, görüntüyü insan gözünün değişiklikleri algılayabileceği şekilde gösterebilmektir. Görsel ataklar daha çok BMP formatındaki dosyalara uygulanabilmektedir.

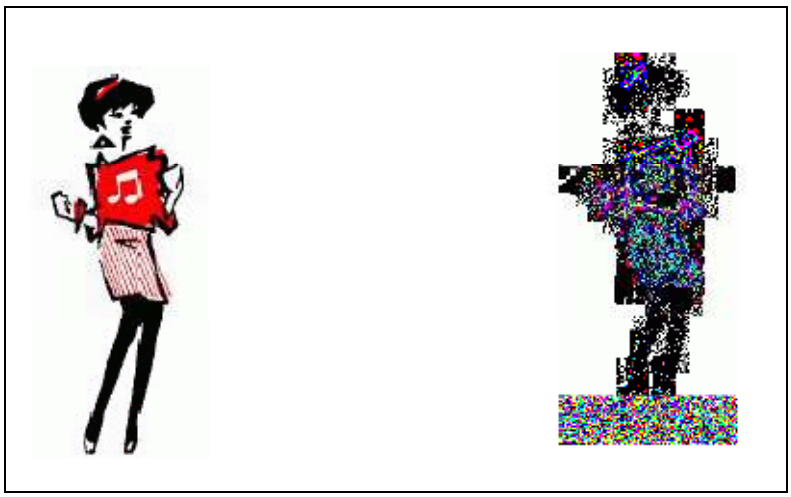
JPEG dosyalar 8x8 piksellik bloklar halinde çalıştığı için bu ataklar sonuç vermemektedir. Görsel ataklarda gizlenen mesaj ya da resmin içindeki bitlerin dağılımı rastgele olmamalıdır.

Şekil 3.1 içinde hiç gizlenmiş veri olmayan bir BMP dosyanın görsel atak ile elde edilmiş görüntüsü vardır:

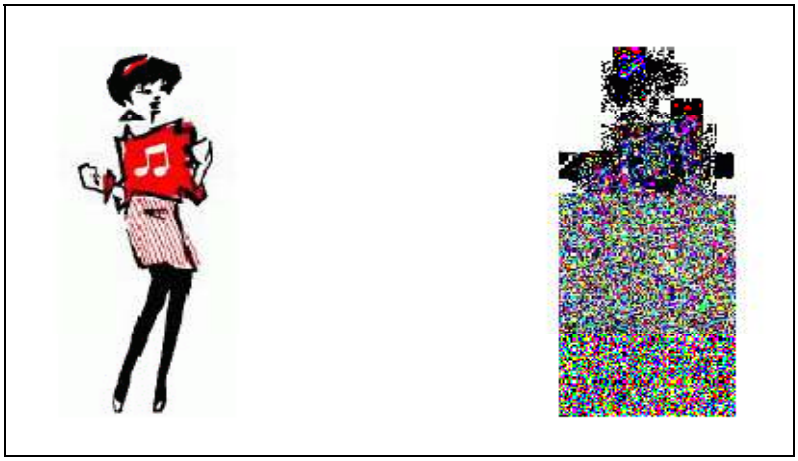


Şekil 3.1. Orijinal resim dosyası ve görsel atak sonucu

Şekil 3.2 ve Şekil 3.3 ise sırasıyla içine 1 Kb 5 Kb büyüklüğünde veri gizlenmiş olan görüntü dosyalarını ve görsel atak sonucu elde edilmiş görüntüleri göstermektedir.



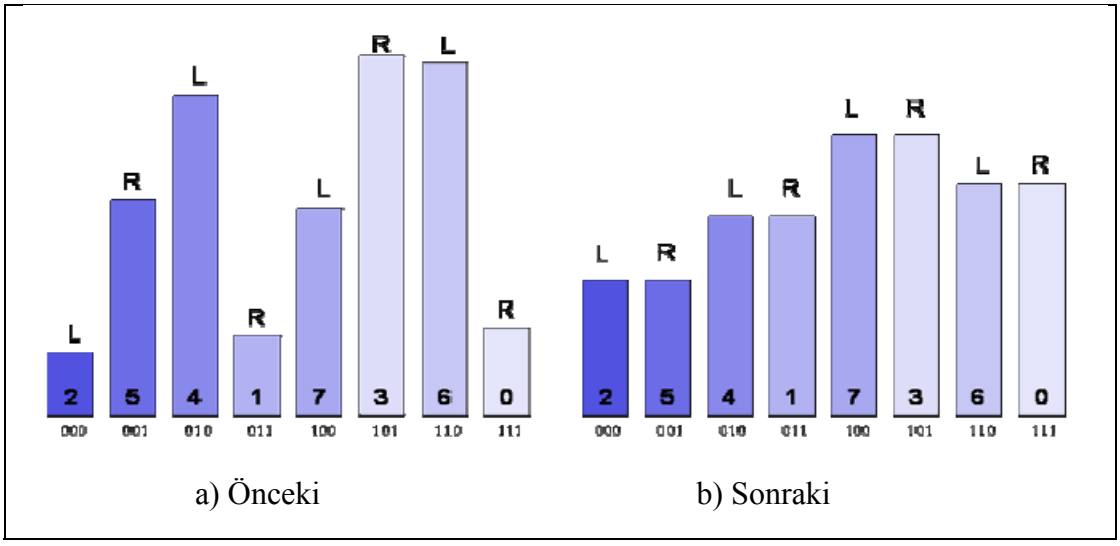
Şekil 3. 2. 1Kb veri gizlenmiş resim dosyası ve görsel atak sonucu



Şekil 3.3. 5Kb veri gizlenmiş resim dosyası ve görsel atak sonucu

3.2. χ^2 Testi

LSB içinde değer çiftleri (DÇ) değerlerinin istatistiksel analizi temelli olan χ^2 istatistik testi, Westfeld ve Pfitzmann tarafından sunulan bir steganaliz metottur [4]. DÇ'ler piksel değerlerinden meydana getirilebilir, ölçülmüş DCT katsayıları veya palet indisleri sadece LSB'de değişmektedir. İçine veri gizlenmemiş görüntüler için DÇ'lerin frekansları düz bir şekilde dağılmamaktadır, fakat LSB gizleme steganografi söz konusu olunca her DÇ' in frekansları eşit olmaktadır.



a) Önceki b) Sonraki

Şekil 3.4. Bir resmin içine gizli mesaj gömülmeden
a) önceki
b) sonraki renk dağılım histogramı.

Bundan sonra Westfeld ve Pfitzmann şüpheli bir görüntüde ölçülen DÇ'lerin oluşunu, istatistiksel rastgele testi ile karşılaştırmıştır. χ^2 istatistik testin ayrıntıları adım adım aşağıda verilmiştir:

Adım 1: *k* kategoriler ve gözlemlerden oluşan rastgele bir örnekleme olduğunu varsayılsın. Her gözlem sadece ve sadece bir kategoriye düşmektedir. Şüpheli bilginin DÇ'lerinin tek değerlerine önem verilmektedir.

Adım 2: Düz bir şekilde dağılmış bir mesajın gizlenmesinden sonra, *i* kategoride teorik olarak beklenen frekansı böyledir:

$$n_i^* = \frac{|\{renk|(renk)'in\ sıralanmış\ indeksi\ \epsilon\ \{2i, 2i+1\}\}|}{2}$$

Adım 3: rastgele örneklemede, ölçülen vuku bulma frekansı aşağıdaki gibidir.

$n_i = |\{renk|(renk)'$in sıralanmış indeksi $= 2i\}|$

Adım 4: χ^2 istatistiği ise k-1 bağımsızlık dereceleri ile şu şekilde hesaplanır:

$$X_{k-1}^2 = \sum_{i=1}^{k} \frac{(n_i - n_i^*)^2}{n_i^*}$$

Adım 5: n_i ve n_i* dağılımları eşit olduğu durumda, *p* mesaj gömme olasılığıdır. Bu olasılık yoğunluk fonksiyonun entegrali alınarak hesaplanmaktadır (Γ , Euler'in gamma fonksiyonudur):

$$P = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma[\frac{k-1}{2}]}\int_0^{x_{k-1}^2} e^{-\frac{x^2}{2}}\ x^{\frac{k-1}{2}-1}\ dx$$

χ^2 istatistiksel analizi sıralı LSB gömme steganografide başarılı sonuç vermiştir. Provos ise bu metodu test aralıkları ve değerleri yeniden değiştirerek genişletmiştir [4]. Provos, test aralığı ve piksel değerlerini *P* ve (*P* +1) piksel çiftinden *P* ve (*P*−1)

çifte kadar yeniden değiştirip, χ^2 testini geliştirmiştir [5].

3.3. Değer Çiftlerinin İstatistiksel Analizi (Histogram analizi)

Pfitzmann ve Westfeld her hangi bir steganografik tekniğe uygulanabilecek güçlü bir istatistiksel saldırı ortaya koydular [58]. Burada sabit Değer Çiftleri (DÇ) kümesi mesaj bitlerini gömmek için birbirinin içine karışmaktadırlar. Örneğin, DÇ piksel değerleri, kuantal DCT katsayıları veya LSB içinde farklılaşan palet indisleri ile oluşturulabilir. Esas görüntüye gömmeden önce, her çiftteki değerlerin meydana çıkması eşitlenme eğimi gösterecektir (bu mesajın uzunluğuna bağlıdır). Bir değeri bir başkasının içine takas etmek görüntüdeki her iki rengin meydana çıkma toplamını değiştirmediğinden, bu keyfiyet istatistiksel Ki-kare testi tasarlamak için kullanılabilir. Her çiftteki her iki değerin meydana çıkmasının aynı olduğu keyfiyetinin istatistiksel önemini test edebiliriz. Eğer, buna ek olarak, stego tekniği mesaj bitlerini takip eden piksellere/indislere/katsayılara sıralı olarak üst sol köşeden başlayarak gömerse, mesajın sonu ile karşılaştığımızda, istatistiksel delilimizde ani bir değişiklik gözlemleyeceğiz.

Örneğin, bir palet görüntüsü için, palette en fazla 256 c_i rengi vardır ki bu en fazla 128 DÇ anlamına gelir. *i*'nci çift, i=1,….,k, için n_1'=1/2 (c_{2i}, c_{2i+1} kümesindeki indislerin sayısı) ve n_i=c_{2i}'ye eşit indislerin sayısı tanımını yaparız. Eğer rastgele bir mesaj gömülmüşse, n_i' teorik olarak beklenen frekans ve n_i c_{2i} renk oluşumunun fiili sayısıdır. Şimdi n_i' ve n_i eşitliği için bir Chi-kare testi yapabiliriz. Ki-kare istatistiği $k–1$ serbestlik derecesi ile

$$x_{k-1}^2 = \sum_{i=1}^{k} \frac{(n_i - n_i')^2}{n_i'}$$

ve n_i' ve n_i dağılımlarının eşit olması olasılığını ifade eden p değeri

$$P = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma[\frac{k-1}{2}]}\int_0^{x_{k-1}^2} e^{-\frac{x^2}{2}}\, x^{\frac{k-1}{2}-1}\, dx$$

olarak hesaplanır.

Sıralı olarak gömülü bir mesaj için, görüntü mesajın gömülü olduğu derecede taranabilir ve zaten uğranılmış bütün piksellerin kümesi için *p* değeri ölçülebilir. *p* değeri başlangıçta 1'e yakın olup sonra mesajın sonuna geldiğimizde birden 0'a düşer. Böylece, bu test yalnız çok büyük olasılıkla bir mesajın gömülü olduğunu değil, ama aynı zamanda gizli mesajın boyutunu da belirlememizi sağlar.

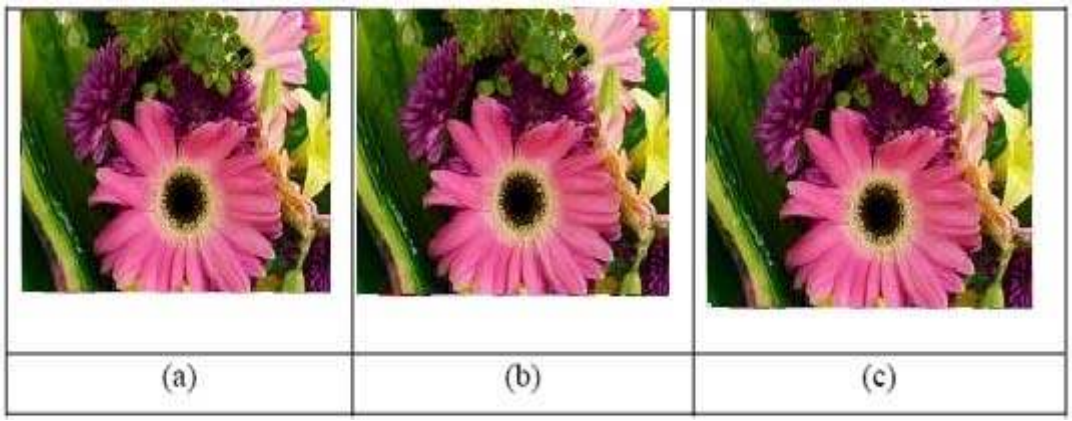
Görüntüdeki mesaj taşıyan pikseller sıra ile değil, rastgele seçilirse, piksellerin çoğunluğu gömme için kullanılmadığı takdirde (görüntüdeki piksel sayısı ile mesaj boyutunun karşılaştırılması), bu test daha az etkili olur. Provos'a göre eğer bu teknik görüntüde daha küçük farklı alanlara uygulanırsa, *p* değeri mesaj uzunluğu arttıkça azalan dalgalanma derecesi ile dalgalanacaktır [58]. Bunun nedeni rastgele yayılmış bir mesajın şansla bazı alanlarda diğerlerine göre daha yoğun olacak olmasıdır. Bu gözlemi sayısallaştırarak, prensip olarak rastgele dağılmış mesajların bile tespit edilebileceğini ve uzunluklarının kestirilebileceğini iddia etmektedir. Maalesef, Provos bu gözlemle ilgili daha çok istatistiksel analiz veya bu iddiayı destekleyen başka detaylar vermemektedir.

Son olarak, örnek sayımları (histogram) analizini esas alan her hangi bir steganalitik tekniğini aldatmanın kolay olacağını belirtiriz. Provos birisinin örneklerin orijinal sayımlarını kendi DÇ'leri içinde koruyacak bir JPEG gömme tekniği (Outguess 0.2) tasarlayabileceğini ve böylece Pfitzmann ve Westfeld'in steganalizi kullanılarak yapılan mesaj tespitinden korunabileceğini göstermektedir.

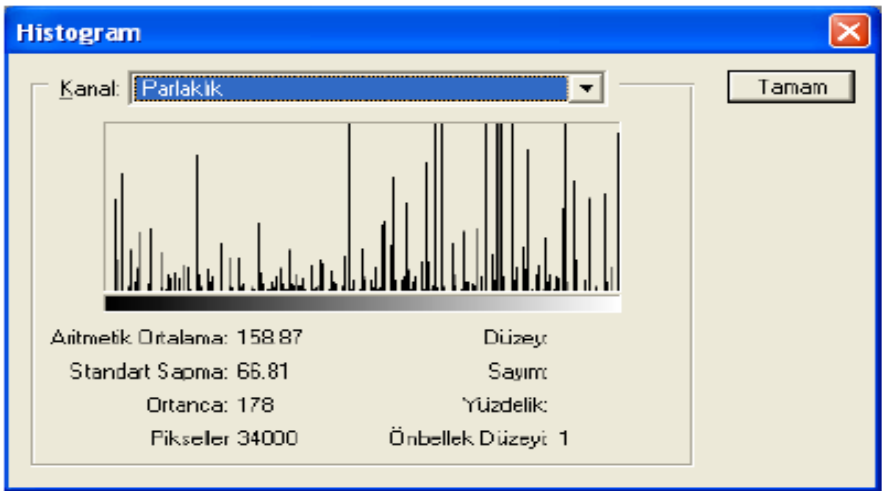
Değer Çiftleri (pairs of values), piksel değerlerinden, ölçülmüş DCT katsayılarından veya LSB'de değişen palet indislerinden oluşturulabilmektedir. Gizlemeden önce taşıyıcı görüntüde her çiftten iki değeri eşit olarak dağılmamaktadır. Mesaj gizlemesinden sonra, her çiftte değerlerin var oluşu eşit olmaya yönelik olacaktır. Bu

analizde χ^2 testi kullanılmaktadır. Her çiftte değerlerin var oluşu eşit olmasının istatistiksel anlamı test edilebilmektedir. Mesela bir palet görüntü için, palet içinde en fazla 256 renk(c_i) vardır, bu demek ki en fazla 128 DÇ vardır.

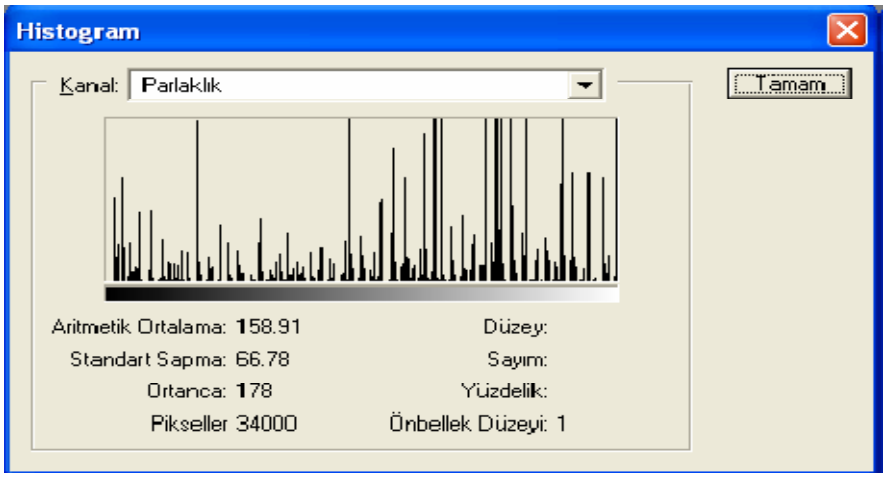
Sıralı bir şekilde saklanmış bir mesaj için, mesajın saklama sırası ile aynen mesajı tarayabilmekte ve bütün piksel değerleri için *p* değeri hesaplanmaktadır. İlk başta *p* 'nin değeri 1'e yakın olacak ve sonra mesajın sonu geldiğinde hemen 0'a düşecektir. Bu en düşük sağ köşesine var olana kadar 0 olarak kalacaktır. Böylece bu test bir mesajın gizleme olasılığını belirtip gizli mesajın büyüklüğünü belirtmektedir. Eğer görüntüde mesaj taşıyıcı pikseller rastgele seçilirse bu test daha az etkili olmaktadır. Provos, bu tekniğin bir görüntünün daha küçük farklı alanlarına uygulanırsa, mesaj uzunluğu arttıkça *p* 'nin değeri gitgide azaldığını fark etmiştir . Fakat Provos bununla ilgili daha ileri istatistiksel analizleri sunmamıştır. Şekil 3.5'de verilen ve içine 1 KB, 5 KB saklanmış olan 200x170 piksel boyutlarındaki resim için histogram sonuçları sırasıyla Şekil 3.6, Şekil 3.7 ve Şekil 3.8'de verilmiştir.



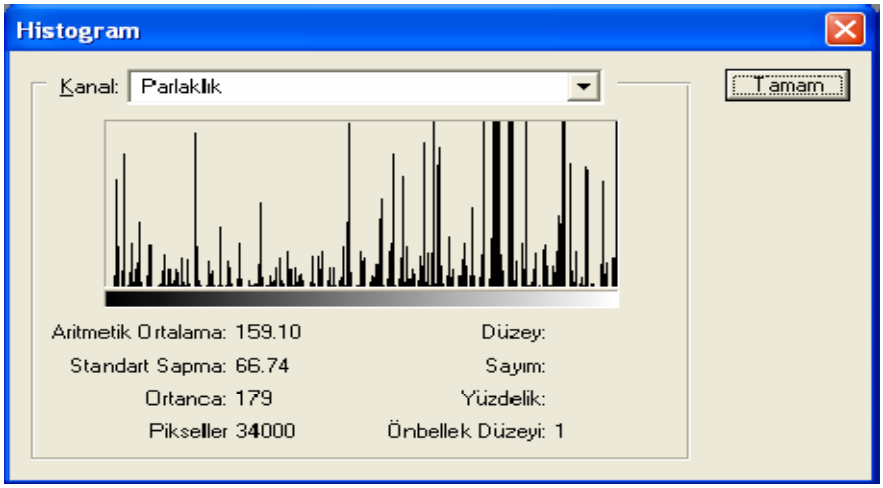
Şekil 3.5. Çiçek resim
a) Orijinal resim (200x170 piksel)
b) 1 KB saklanmış resim
c) 5 KB saklanmış resim



Şekil 3.6. Orijinal resim için histogram



Şekil 3.7. 1KB veri saklanmış resmin histogramı



Şekil 3.8. 5KB veri saklanmış resmin histogramı

Resmin içine saklanan veri miktarı arttıkça histogramda oluşan değişiklikler oldukça fark edilir hale gelmektedir.

3.4. İkili İstatistik Metotları (RS Steganaliz)

Bu analiz, görüntülerde uzaysal ilgileşimlerden üretilen duyarlı ikili istatistiklerini kullanmaktadır. RS steganalizi 24 bit renkli ve 8 bit gri seviye görüntülerde kullanılmaktadır. RS steganalizinde, bir görüntünün piksellerinin 3 bağımsız gruba: Düzenli (Regular-R), Tekil (Singular-S) ve Kullanılmayan (Unused-U) olarak ayrılması esastır. Fridrich ve Goljan tarafından geliştirilmiştir [92]. Sayısal kamera veya tarayıcı ile alınan yüksek kaliteli görüntüler için, RS steganalizi güvenli bit-oranın her örnekleme için 0,005 bitten küçük olduğunu göstermektedir. RS' yi açıklamak için birkaç nütasyona ihtiyaç duyulmaktadır. Test edilen görüntü (*R*) , *P* kümesinden değer alan *MxN* pikselden oluşmaktadır. Örnek olarak, 8-bit gri seviyeli bir görüntüde, *P* = {0,....,255}'tir. Sonra *R* , *n* komşu pikselden oluşan *G* ayrı gruba bölünmektedir:

$$G = (x_1,.....,x_n) \in R \tag{3.1}$$

Ayrıcı fonksiyon aşağıdaki gibi belirlenmiştir:

$$f(x_1, \dots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \tag{3.2}$$

Genelde *G* ne kadar büyükse, *f* fonksiyonun değeri o kadar büyük olmaktadır. *P* 'de ters işlemi *F* (flip-ping), aşağıdaki gibi belirlenmiştir:

$$F_i(F_i(x)) = F_0(x) = x , \quad i \in \{-1,1\}$$
$$F_1 : 0 \leftrightarrow 1 , 2 \leftrightarrow 3 ,....., 254 \leftrightarrow 255 \tag{3.3}$$
$$F_{-1} : -1 \leftrightarrow 0 , 1 \leftrightarrow 2 ,....., 253 \leftrightarrow 254 , 255 \leftrightarrow 256$$
$$F\text{-}1(x) = F1(x+1) - 1 \quad \text{her x için}$$

Sonra *G* grubu 3 çeşit piksel gruplarda tanımlamaktadır:

Düzenli gruplar: $G \in R \leftrightarrow f(F(G)) > f(G)$

Tekil gruplar: $G \in S \leftrightarrow f(F(G)) < f(G)$

Kullanılmayan gruplar: $G \in U \leftrightarrow f(F(G)) = f(G)$ Her maske için M, $F_{M(1)}(X_1),.....,F_{M(n)}(X_n))$, R (Regular-Düzenli), S (Single-Tekli), U (Unused-Kullanılmayan) çeşitlerinden birinde tanımlanır. Fridrich ve Goljan [92] işlenmemiş yeni BMP, JPEG ve işlenmiş BMP görüntülerden oluşan büyük veritabanları için 2 istatistiksel tahminleri deneysel olarak ispatlamıştır [92]:

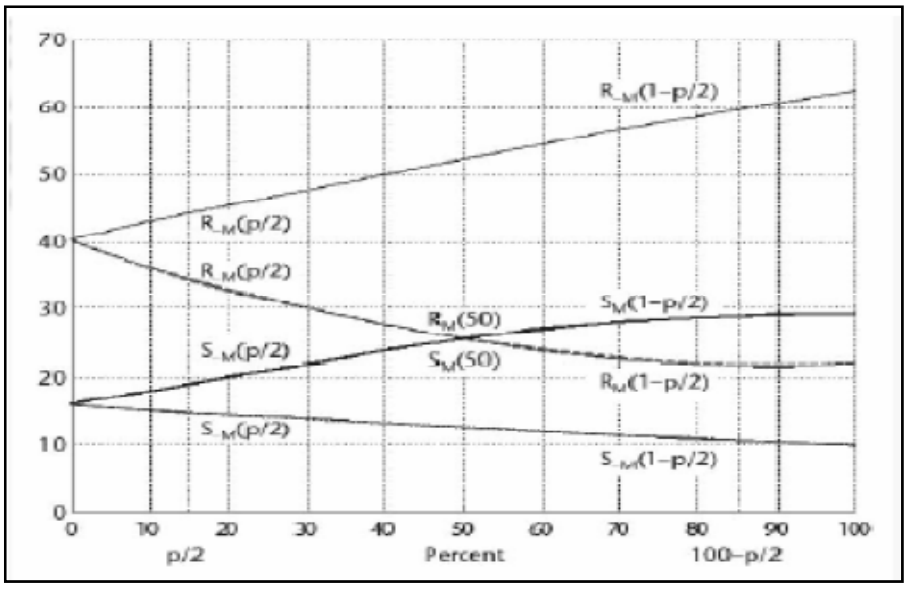
$$R_M + S_M \leq 1 \quad \text{ve} \quad R_{-M} + S_{-M} \leq 1$$

$$R_M \cong R_{-M} \quad \text{ve} \quad S_M \cong S_{-M} \tag{3.4}$$

$$R_M(50) = S_M(50) \tag{3.5}$$

M maskesi $M = [F_0\ F_1;\ F_1\ F_0]$ göstermekte ve $-M$ de $[\ F0\ F\text{-}1;\ F\text{-}1\ F0]$'ı göstermektedir.

Birçok deneyden sonra Fridrich RS-diyagram hakkında bir değerlendirme elde etmiştir.



Şekil 3.9. RS diyagramı

p uzunluklu bir mesaj bir stego görüntü içine gömülmektedir. Mesajın, stego görüntü piksellerin rastgele bit akımı olduğu farz edilip pikseller kendilerine karşı olanlara gelmektedir. 1 *F* kaydırma işlemi ve −1 *F* kaydırma işlemi bütün piksellere uygulanarak aşağıdaki 4 nokta hesaplanmaktadır:

$R_M(1-p/2), S_M(1-p/2), R_{-M}(1-p/2), S_{-M}(1-p/2)$

Gizli mesajın uzunluğu için, *p Eş*.3.4 ve *Eş*. 3.5 varsayımlardan basit bir Eşitlik Eş. 3.6 üretmek mümkündür.

$$2(d1 + d0)\,x^2 + (d_{-0} - d_{-1} - d_1 - 3d_0)\,x + d_0 - d_{-0} = 0 \qquad (3.6)$$

$d_0 = R_M(p/2) - S_M(p/2)$
$d_1 = R_M(1-p/2) - S_M(1-p/2)$
$d_{-0} = R_{-M}(p/2) - S_{-M}(p/2)$
$d_{-1} = R_{-M}(1-p/2) - S_{-M}(1-p/2)$
Mesaj uzunluğu (*p*), *x* kökünden hesaplanmaktadır:

$$P = x / (x - ½) \qquad (3.7)$$

Eğer bir stego görüntü Eş. 3.6 eşitliğinin sabit katsayısının 0 olmasını sağlıyorsa Eş.3.4 varsayım bu görüntünün mesaj uzunluğu (*p*) da 0 olarak değerlendirilir.

3.4.1. İkili İstatistik Metotları (RS Steganaliz) örnek uygulama

Westfeld ve Provos metotları gibi örnek sayımlardan yola çıkan istatistiksel metotlar büyük miktarda veya çok önemli bilgiyi ihmal etmektedirler (piksellerin stego görüntüye yerleştirilmesi). Stego görüntüdeki uzaysal ilgileşimler kullanılarak, çok daha güvenilir ve doğru tespit yapılabileceği sezgisel olarak açıktır. Bununla birlikte, görüntüde mevcut bazı sözde rastgele bileşenler ile görüntünün kendisi arasındaki zayıf ilişkiyi açığa çıkarmak ve sayısallaştırmak kolay değildir (örneğin LSB düzlemi). Bir ölçü kullanılarak bu ilişki bir kez sayısallaştırılırsa, bu ölçünün mesaj gömme ile nasıl değiştiği incelenebilir. Türetilen ilişki steganalitik teknikler için

temel olarak hizmet görebilir. Aşağıda, yukarıda sunulan fikirlerin uzaysal etki alanında gömülü LSB için nasıl gerçekleştirilebileceğini gösterilmektedir.

MxN pikselli bir üst görüntümüz olduğunu ve piksel değerlerinin bir *P* kümesinden alındığını kabul edelim. Örneğin, 8 bitlik bir gri ölçekli görüntü için, *P*={0,…,255}. Uzaysal ilgileşimler $G=(x_1,...,x_n)$ piksel grubuna $f(x_1,...,x_n)\varepsilon R$ gerçek sayısını atayan bir *f* fark fonksiyonu kullanarak yakalanmaktadır. Bu çalışmada, aşağıdaki gibi tanımlanan *f* fonksiyonu kullanılmaktadır:

$$f(\mathrm{x}_1, \dots, x_n) = \sum_{i=1}^{n-1} |\mathrm{x}_{\mathrm{i}+1} - \mathrm{x}_{\mathrm{i}}| \tag{3.8}$$

Bu *G*'nin düzgünlüğünü ölçmektedir – *G* grubu ne kadar gürültülü ise, fark fonksiyonunun değeri o kadar büyüktür.

LSB gömülmesi görüntüdeki gürültüyü arttırmaktadır, bu yüzden *f*'nin değerinin LSB gömülmesinden sonra artması beklenmektedir. LSB gömme işlemi bir F_1:0↔1, 2↔3,….,254↔255 çevirme fonksiyonu kullanılarak kolayca tarif edilebilir. *x* gri seviyesinin LSB'sini değiştirmek *F*'yi *x*'e çevirmekle aynıdır. Ayrıca F_{-1}'i -1↔0, 1↔2, 3↔4,…,253↔254, 255↔256 şeklinde çevirerek, veya

$$\mathrm{F}_{-1}(\mathrm{x}) = \mathrm{F}_1(\mathrm{x}+1) - 1 \ \forall x. \tag{3.9}$$

Kaydırılmış LSB diye adlandırılan ikili bir kavram tanımlanmaktadır.

Son olarak, F_0'da $F(x)=x, \ \forall x \varepsilon P$ özdeşlik permütasyonu olarak tanımlanmaktadır. *f* fark fonksiyonu ve *F* çevirme operatörü üç tip piksel grubu tanımlar: çevirmenin fark fonksiyonunun değerini nasıl değiştirdiğine dayanarak *R,S* ve *U*: eğer $f(F(G))>f(G)$ ise *G* düzenlidir, eğer $f(F(G))<f(G)$ ise *G* tekildir ve eğer $f(F(G))=f(G)$ ise *G* değişmez. Burada, *F(G)* *F* çevirme fonksiyonunu $G=(x_1,...,x_n)$ vektörünün bileşenlerine uygulamak anlamına gelmektedir.

Genel olarak, G grubundaki değişik piksellere değişik çevirme uygulamaları istenmektedir. Piksellere çevirme atamasını -1,0 ve 1 değerlerine sahip bir n-değişken seti olan M maskesi ile yakalayabilir. $F(G)$ çevrilmiş grubu $(F_{M(1)}(x_1), F_{M(2)}(x_2),\ldots,F_{M(n)}(x_n))$ olarak tanımlanmaktadır.

Tipik görüntülerde, G grubunu çevirmek f fark fonksiyonlarında düşüşten daha çok artışa yol açacaktır. Bu yüzden, düzenli grupların sayısı tekil grupların sayısından daha çok olacaktır. Negatif olmayan M maskesi için düzenli grupların nispi sayısını R_M (bütün grupların yüzdesi olarak) adlandırılır ve S_M tekil grupların nispi sayısı olsun. Böylece $R_M+S_M\leq 1$ ve $R_{-M}+S_{-M}\leq 1$ olur. Steganalitik metodu sıfır mesaj hipotezi tipik üst görüntüler içindir, R_M değeri yaklaşık olarak R_{-M} değerine eşittir ve aynı şey S_M ve S_{-M} için de doğru olmalıdır:

$$R_M \cong R_{-M} \; and \; S_M \cong S_{-M} \tag{3.10}$$

Bu hipotez Denklem (3.9) incelenerek buluşsal olarak ispat edilebilir. Çevirme operasyonunu kullanarak, F_{-1} F_1'i renkleri bir birim kaydırılmış görüntüye uygulamakla aynıdır. f fark fonksiyonu düzgünlüğü yakaladığı için, bütün piksellere 1 değeri eklemek düzenli ve tekil grupların istatistiğini hiçbir şekilde etkilemez. Kuşkusuz, Denklem (3.10)'ün hem JPEG hem sıkıştırılmamış formatlarda bir dijital kamera ile alınmış görüntüler için çok doğru olduğunu gösteren yaygın deneysel sonuçlar vardır. Bu aynı zamanda genel görüntü işlem operasyonları ve taranmış fotoğraflar için de geçerlidir. Bununla birlikte, Denklem (3.10)'daki ilişki LSB düzlemi rastgele hale getirildikten sonra (örneğin LSB steganografi nedeniyle) ihlal edilmektedir. LSB düzlemi kuvvetlerinin rastgele hale getirilmesi gömülü mesajın m uzunluğu arttıkça R_M ve S_M arasındaki farkı sıfıra götürür. Piksellerin 50%'sinin LSB'sini çevirdikten sonra (bu her pikselin içine bir rastgele mesaj biti gömdükten sonra ne olacağıdır), $R_M \cong S_M$ elde edilir. Esas şaşırtıcı olan LSB düzlemini rastgele hale getirmenin R_{-M} ve S_{-M} üzerinde *ters* etki yapmasıdır. Bunların farkı gömülü mesajın m boyu uzadıkça artmasıdır. R_M, S_M, R_{-M}, S_{-M}'yi çevrilmiş LSB'li piksel sayısının fonksiyonu olarak gösteren grafik Şekil 3.10'da görülmektedir (RS diyagramı). Diyagram aslında R_M ve S_M'nin bütün muhtemel LSB rastgelelik

hallerinin istatistiksel örneği üzerinden beklenen değerleri göstermektedir, ancak, nütasyonu basitleştirmek için, beklenen değerler için aynı simgeler kullanmaktadır.

Çizelge 3.1. Dört klik tipi.

Klik tipi	F_1 çevirme	F_{-1} çevirme
$r=s=t$	2R, 2S, 4U	8R
$r=s>t$	2R, 2S, 4U	4R, 4U
$r<s>t$	4R, 4S	4R, 4S
$r>s>t$	8U	8U

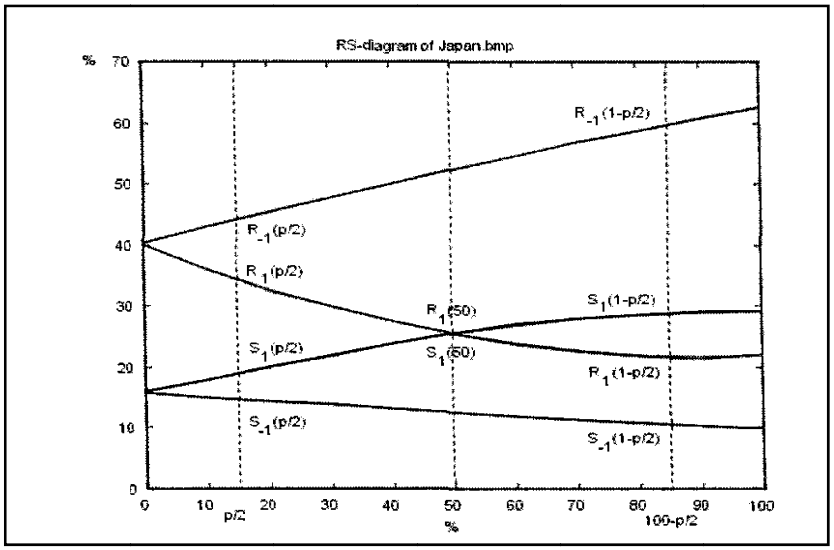
M=[0 1 0] maskesi için R_{-M} ile S_{-M} arasındaki farktaki garip artış için basit bir açıklamak bulunmaktadır. Diğer maskeler için de benzer deliller bulunabilir. $C_i=\{2i, 2i+1\}$, i=0,…,127 kümelerini ve $C_{rst}=\{G/G\varepsilon C_r x C_s x C_t\}$ gruplarının kliklerini tanımlarız. Her biri 8 gruptan (üçlü) oluşan 128^3 klik vardır. Klikler LSB rastgeleliği kapsamında kapanır. Analiz amacı için, yatay ve düşey simetrik klikler ihmal edilerek dört farklı klik tipi oluşturulur. Çizelge 3.1 bu dört klik tipini ve rastgelelilik sonrası F_1 ve F_{-1} kapsamındaki R,S ve U gruplarının sayısını göstermektedir. Çizelgeden LSB' lerin rastgeleliğinin F_1 kapsamındaki her klikte R ve S gruplarının sayısını eşitleme eğilimine karşın, F_{-1} kapsamında R gruplarının sayısını arttıracağı ve S gruplarının sayısını azaltacağı görülebilir.

RS Steganalizi olarak adlandırılan yeni steganalitik metodun prensibi RS diyagramının dört eğrisini kestirmek ve ekstrapolasyon ile kesişmelerini hesaplamaktır. R_{-M} ve S_{-M} eğrilerinin düz çizgilerle iyi modellendiğini, buna karşılık ikinci derece polinomların "iç" R_M ve S_M eğrilerini bir hayli iyi yaklaştırdığını gösteren deneysel sonuçlar vardır. Eğrilerin parametrelerini Şekil 3.10'de işaretlenen noktalardan belirlenebilir.

Eğer rastgele dağılmış piksellerin LSB'lerinde gömülü p bilinmeyen uzunluğunda (piksellerin yüzdesi olarak) bir mesajın stego görüntüsüne sahipsek, R ve S gruplarının sayısına ait başlangıç ölçümümüz $R_M(p/2)$, $S_M(p/2)$, $R_{-M}(p/2)$ ve $S_{-M}(p/2)$ noktalarına mütekabil olur (bak Şekil 10). Faktör yarımdır çünkü – mesajın rastgele bit akıntısı olduğu varsayılarak – piksellerin ortalama yalnız yarısı mesaj gömmeyle

çevrilecektir. Görüntüdeki bütün piksellerin LSB'lerini çevirir ve *R* ve *S* gruplarının sayısı hesaplaranak, dört $R_M(1-p/2)$, $S_M(1-p/2)$, $R_{-M}(1-p/2)$, $S_{-M}(1-p/2)$ noktasını elde edilir.

Dijital kamera ile alınan bir görüntünün Şekil 3.10'da RS diyagramı. x-ekseni çevrilmiş LSB'li piksellerin yüzdesi ve y-ekseni M ve –M maskesi, M=[0 1 1 0], olan düzenli ve tekil grupların nispi sayısıdır.



Şekil 3.10. Dijital kamera ile alınan bir görüntünün RS diyagramı

Stego görüntünün LSB düzlemini rastgele hale getirerek, $R_M(1/2)$ ve $S_M(1/2)$ orta noktaları elde edilmektedir. Bu iki nokta LSB' lerin özel rastgeleliğine dayandığı için, işlemi birçok kez tekrarlayıp istatistiksel örneklerden $R_M(1/2)$ ve $S_M(1/2)$ kestirilmektedir. Bu zaman alıcı istatistiksel kestirimden kurtulmak ve aynı zamanda mesaj uzunluğu kestirimini çok daha mükemmel yapmak iki tane daha (doğal) varsayımı kabul etmekle mümkündür. Bunlar:

1. R_M ve R_{-M} eğrilerinin kesişme noktası S_M ve S_{-M} eğrilerinin kesişme noktası ile aynı *x* koordinatına sahiptir (bu aslında Denklem 3.10'tür)

2. $R_M(1/2)= S_M(1/2)$.

Birinci varsayımı işlenmemiş ham BMP'ler, JPEG'lere sahip görüntüler ve işlenmiş görüntülere ait büyük bir veri tabanı için deneysel olarak ispat edildi (ikinci varsayım da gerçekten ispatlanabilir).

p/2 ve 1-*p*/2'de *R* ve *S* gruplarının sayısı doğru çizgileri tanımlar, ve geriye kalan noktalar yukarıdaki varsayım (3.8) ve (3.9) ile birlikte parabolleri ve kesişmelerini ayrı ayrı belirleyecek yeterli kısıtlamayı sağlarlar. Doğrusal değiştirme *z=(x-p/2)/(1 p)* ile elde edilen *p/2* 0 ve 100- p/2 1 olacak şekilde *x* eksenini yeniden ölçeklendirdikten sonra, kesişme noktasının *x* koordinatı aşağıdaki ikinci derece denkleminin kökünden hesaplanabilir:

$d_0 = R_M (p / 2) - S_M (p / 2)$, $d_1 = R_M (1- p / 2) - S_M (1- p / 2)$, $d_{-0} = R_{-M} (p / 2) - S_{-M} (p / 2)$, $d_{-1} = R_{-M} (1- p / 2) - S_{-M} (1- p / 2)$

ise

$2(d1 + d0) z^2 + (d_{-0} – d_{-1} – d_1 – 3d_0) z + d_0 – d_{-0} = 0$

Mutlak değeri

$$P = z / (z – ½) \tag{3.11}$$

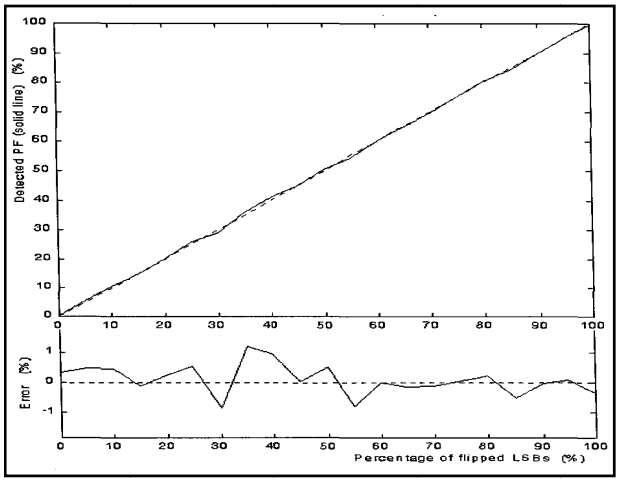
değerinden küçük olan *z* kökünden mesaj uzunluğu *p*'yi hesaplanır. Hesaplanan mesaj uzunluğunun doğruluğunu etkileyen üç ana faktör vardır. Bunlar başlangıçtaki hata, gürültü seviyesi veya üst görüntünün kalitesi ve mesaj bitlerinin görüntüye yerleştirilmesidir.

Başlangıçtaki hata: Orijinal üst görüntü için bile rastgele değişkenlikler nedeniyle RS Steganalizi sıfır olmayan küçük bir mesaj uzunluğu belirtebilir. Başlangıçtaki bu sıfır olmayan hata pozitif veya negatif olabilir ve RS Steganalizinin elde edilebilir doğruluğuna sınır getirir. Bu başlangıç hatası 331 gri ölçekli JPEG görüntüsünün büyük veri tabanı için test edilmiş ve 0,5% standart sapmalı bir Gauss dağılımı bulunmuştur (toplam 1 bpp kapasitenin yüzdesi olarak ifade edilen mesaj uzunluğu). Daha küçük görüntülerde *R* ve *S* gruplarının daha küçük olması nedeniyle başlangıç hatası için daha büyük değişkenlikler gösterme eğilimi vardır. Yarım tonlu

görüntüler ve gürültü görüntülerinin taranması aynı şekilde daha büyük değişkenlikler göstermektedir. Öte yandan, JPEG görüntüleri, dijital kamera tarafından elde edilen sıkıştırılmamış görüntüler, fotoğraf taramaları ve tipik görüntü işleme filtreleri ile işlenmiş görüntüler için hata tipik olarak çok düşüktür. Bir başka başparmak kuralı olarak, renkli görüntülerin başlangıç hatasında gri ölçekli olanlara nazaran daha büyük değişkenlikler gösterdiği ifade edilebilir.

Gürültü: Çok gürültülü görüntüler için, üst görüntüdeki düzenli ve tekil piksel sayıları arasındaki fark küçüktür. Dolayısıyla, RS diyagramındaki çizgiler küçük bir açı ile kesişirler ve RS Steganalizinin doğruluğu azalır. Aynı şey düşük kaliteli görüntüler, çok sıkıştırılmış görüntüler ve küçük görüntüler için de doğrudur (yetersiz istatistik nedeniyle).

Mesaj yerleştirilmesi: RS Steganalizi stego görüntü içine rastgele dağıtılmış mesajlarda görüntünün lokal bir bölgesinde yoğunlaşmış mesajlara nazaran daha doğrudur. Bu soruna hitap etmek için aynı algoritma görüntünün kayan dikdörtgen bir bölgesine uygulanabilir.



Şekil 3.11. ‘Kyoto.BMP’ için, RS Steganalizi diyagramı

Şekil 3.11’de ‘Kyoto.BMP’ için, RS Steganalizi kullanılarak çevrilmiş piksellerin kestirilmiş yüzdesine(sürekli çizgi) karşı çevrilmiş piksellerin gerçek sayısı. Şeklin alt kısmı büyütülmüş tespit hatasını göstermektedir.

Şekil 3.11'de RS Steganalizin başarılı olduğunu göstermektedir. Kodak DC260 dijital kamera ile renkli 1536x1024 görüntü 'Kyoto.BMP' alınmış, gri ölçeğe dönüştürülmüş, ve 384x256 piksellik bir örneği çıkartılmıştır. 0-100% pikselin 5% artışlarla LSB'leri rastgele hale getirilerek bir dizi stego görüntü yaratılmıştır (pikseller rastgele seçilmiştir). Çevrilmiş piksellerin gerçek ve kestirilmiş yüzdeleri arasındaki hata hemen her zaman toplam görüntü kapasitesinin 1%'inden daha azdır.

RS Steganalizi ticari steganografik yazılım ürünlerinin çoğunda uygulanabilir. Saldırıya açık programların örneklerine örneğin, Steganos, Windstorm, S-Tools, ve Hide4PGP dâhildir. Wbstego ve Encrypt Pic LSB gömmeyi sıralı piksellerin içine koyarlar, bu yüzden onları analiz etmek için Kısım 3.3'nin metodunu kullanmak daha iyidir. RS steganalitik metodu bu yazılım ürünleri tarafından işleme tabi tutulan ve değişik mesaj boyutu içeren az miktar görüntü örneği üzerinde test edilmiştir. Her durumda, metot stego görüntüleri orijinal üst görüntülerden kolayca ayırt etti ve kestirilmiş mesaj uzunluğu toplam görüntü kapasitesinin çok küçük bir yüzdesi oldu. RS Steganalizin performansının günümüz steganografik yazılımları kullanılarak elde edilen görüntüler üzerinde test edilmesi için, kısa mesajlı nispeten küçük bir görüntü kullanılmıştır. Kodak DC260 dijital kamera tarafından alınan (orijinal çözümleme 1536x1024) ve JPEG formatında orijinal olarak saklanan 24-bit renkli fotoğraf 1024x744 piksele kırpıldı. Toplam görüntü kapasitesinin 5%'i uzunluğunda bir mesaj (100%=piksel başına 3 bit) rastgele seçilen piksellerin LSB'sine gömüldü. Sonuçlar Çizelge 3.2'de gösterilmiştir, ve ideal durumda tespit edilmesi gereken gerçek sayılar (sıfır hata varsayımı) parantezler içinde belirtilmiştir.

Çizelge 3.2 . Test görüntüsü 'cat.BMP' için çevrilmiş LSB'li piksellerin başlangıç hatası ve kestirilmiş sayısı.

Görüntü	**Kırmızı (%)**	**Yeşil (%)**	**Mavi (%)**
0,00 Başlangıç hatası	0,00 (0,00)	0,17 (0,00)	0,33 (0,00)
Steganos	2,41 (2,44)	2,70 (2,46)	2,78 (2,49)
S-Tools	2,45 (2,45)	2,62 (2,43)	2,75 (2,44)
Hide4PGP	2,44 (2,46)	2,70 (2,46)	2,85 (2,45)

RS steganalizinin prensipleri palet görüntülerinin indisleri içindeki LSB gömülüşünün varyantları ve JPEG dosyalarındaki sayıya dökülmüş DCT Katsayılarına genişletilebilir.

(a) (b)

Şekil 3.12. Test görüntüsü
a) Kyoto.BMP
b) cat.BMP

3.5. JPEG Steganaliz

JPEG Steganaliz JPEG formatındaki dosyalar üzerinde uygulanan steganalitik yöntemlerdir. JPEG dosyalarındaki steganalizde iki durum vardır [40,41].

- Hem orijinal hem de içine bilgi saklanmış resmin elimizde olduğu durum (Bilinen stego saldırısı).
- Sadece bilgi saklanmış resmin elimizde olduğu durum (Seçilmiş stego saldırısı).

Orijinal ve bilgi gizli resmin elimizde olduğu durum:

Bu durumda her iki resmin ilk blokları için Nicelendirilmiş DCT matrisleri bulunur ve aralarındaki farka bakılır.

Bu durumu açıklamak için bir örnek ele alalım [40]. Şekil 3.13'da orijinal resim ve içinde 1,5 KB gizli bilgi bulunan resim verilmektedir. Aslında JPEG dosya formatından dolayı dosya büyüklüklerinde farklılık meydana gelmektedir. Buradan dosyanın içinde bilgi gizli olduğu anlaşılmaktadır. Bir sonraki adım bu bilgiyi elde etmek olacaktır.

(a)

(b)

Şekil 3.13. Orijinal ve içinde bilgi saklı örnek resim dosyaları
a) Orijinal resim (16778 Byte)
b) İçinde bilgi saklı resim (17344)

Şimdi elimizdeki 8x8 piksellik blok içinde bilgi olduğunu varsayalım. Bu bloğun; orijinal ve içinde bilgi gizli olan resim için nicelendirilmiş DCT katsayı matrisi Şekil 3.14'deki gibidir. Bu matris ilk 8x8'lik blok için verilmiştir (piksel değerleri hexadecimal olarak verilmiştir).

Orijinal resmin ilk 8x8'lik bloğu							
D6	69	13	05	03	15	F2	EB
FF	04	01	00	FA	FB	F9	FF
06	02	FE	FF	00	00	00	FF
01	03	02	01	01	FF	00	00
01	00	00	00	00	00	00	00
00	FF	FF	00	00	00	00	00
00	00	00	00	00	01	00	00
00	00	00	00	00	00	00	00

İçinde bilgi saklı resmin ilk 8x8'lik bloğu							
D6	69	12	05	03	15	F3	EA
FE	04	01	00	FA	FB	F8	FE
06	03	FE	FF	00	00	00	FE
01	02	03	01	01	FE	00	00
01	00	00	00	00	00	00	00
00	FE	FF	00	00	00	00	00
00	00	00	00	00	01	00	00
00	00	00	00	00	00	00	00

Şekil 3.14. Orijinal ve içinde bilgi saklı örnek resimim onaltılık sisteminin görüntüsü

Daha sonra içinde bilgi saklı resmin ilk 8x8 piksellik bloğunun son bitlerini gösteren matris yazılır.

Orijinal resmin nicelendirilmiş DCT katsayı matrisinde 00 ve 01 olan değerlerin olduğu pikseller bit alımında kullanılmamaktadır. O yüzden bilgi gizlenmiş matriste

koyu renkle gösterilmiştir ve Şekil 3.15'deki hesaplamalarda da göz önüne alınmamaktadırlar.

$$\begin{bmatrix} 00 & 01 & 00 & 01 & 01 & 01 & 01 & 00 \\ 00 & 00 & \mathbf{01} & \mathbf{00} & 00 & 01 & 00 & 00 \\ 00 & 01 & 00 & 01 & \mathbf{00} & \mathbf{00} & \mathbf{00} & 00 \\ \mathbf{01} & 00 & 01 & \mathbf{01} & \mathbf{01} & 00 & \mathbf{00} & \mathbf{00} \\ \mathbf{01} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} \\ \mathbf{00} & 00 & 01 & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} \\ \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{01} & \mathbf{00} & \mathbf{00} \\ \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} \end{bmatrix}$$

Şekil 3.15. Orijinal resmin nicelendirilmiş DCT katsayı matrisinde

İçinde bilgi olan resmin ilk 8x8 piksellik bloğu şunları içermektedir:

- İlk 5 bit kendinden sonraki kaç bitin mesaj uzunluğunu belirlemek için alınacağını belirler. Yani:
  - 00 01 00 01 01 ⇒ 01011=11, Bu da ilk beş bitten sonraki 11 bit mesajın uzunluğunu vereceğini belirtir.
- Daha sonraki 11 bit'ten mesajın uzunluğunu bulunur.
  - 01 01 00 00 00 00 01 00 00 00 01⇒ 11000010001=1553
- Sonra gelen 8 bit ise mesajın ilk karakterini vermektedir.
  - 00 01 00 00 01 00 00 01 ⇒ 01001001=73 (Büyük harf I)

Mesajımız (I harfi) ile başlamaktadır. Bu şekilde 1553 adet blok incelenerek tüm mesaj elde edilebilir.

Diğer bloklarda işlem yapılırken ilk iki adım uygulanmaz ve doğrudan mesajın karakterini elde etme işlemine geçilir.

Sadece bilgi gizli resmin elimizde olduğu durum:

İkinci durumda elimizde sadece şifrelenmiş resim vardır. Bu durumda hangi bloklarda şifrelenmiş metin olduğunu anlamak amacı ile JPEG Uygunluk Esasına Dayanan Steganaliz yapılır [92].

Eğer bir taşıyıcı-görüntü ilk olarak JPEG formatında saklanmışsa, JPEG sıkıştırma tarafından yaratılan yapının özellikleri mesaj gizlemeden dolayı silinmeyecektir, sadece biraz değişecektir.

Stego-görüntüden 8x8 piksellik bloklardaki DCT katsayıların değerlerini analiz ederek JPEG ölçme Çizelgesi elde edilebilmektedir.
Bir görüntüdeki hangi 8x8 piksellik blok JPEG sıkıştırma ile uygunluk göstermiyorsa, bundan gizlenen mesajın uzunluğu ve yerleşimi bulunmaktadır.
Algoritmanın adımları aşağıdaki şekildedir.

Adım 1: Resmi 8x8 piksellik bloklara böl. Eğer resmin boyutu 8'in katı olarak değilse son birkaç satır ya da sütun yok sayılabilir.

Adım 2: Listeden tüm doymuş blokları çıkartarak blokları yeniden düzenle(en az bir pikseli 0 ya da 255 değerinde ise o blok doymuş demektir.). Blokların toplam sayısını T olarak belirle.

Adım 3: Tüm T bloklarının Q nicelendirme (quantization) matrisini çıkar. Eğer Q'nun tüm elemanları aynı ise bu resim JPEG olarak kaydedilmemiştir ve bu steganaliz yöntemi uygulanamaz (algoritmadan çık). Q için bir ya da daha fazla makul sonuç mevcut ise devam et.

Adım 4: Her B bloğu için S niceliğini aşağıdaki formülü kullanarak hesapla;

$$S = \sum_{i=1}^{64} |QD'(i) - q_{p(i)}(i)|$$

i. Eğer $S > 16$ ise B bloğu Q nicelendirme matrisi ile JPEG sıkıştırmasına uyumlu değildir.

ii. Eğer $S \leq 16$ ise her QD_i DCT katsayısı için $Q(i)$ 'lerin en yakın katlarını

hesapla ve ' QD_i 'den uzaklıklarına göre sırala, ve bunları $q_{p(i)}$, p=1,… şeklinde belirt.

iii. $S = \sum_{i=1}^{64} |QD'(i) - q_{p(i)}\ (i)| \leq 6$

eşitsizliğini sağlayan tüm kombinasyonlar için şu eşitliği kontrol et:

$B = |DCT^{-1}\ (QD)|, \quad QD(i) = q_{p(i)}\ (i)$ ise

iv. Eğer $\{p(1),.....,p(64)\}$ sıralanmış kümesinin en az biri için bu denklem sağlanırsa, B bloğu JPEG uyumludur aksi durumda değildir.

Tüm bloklarının analizinden sonra, uyumsuz JPEG bloğu bulunmaz ise bu resmin içinde gizli bilgi yok demektir. Diğer yandan eğer birkaç tane uyumsuz JPEG bloğu var ise gizli mesaj vardır.

3.5.1. JPEG uyumluluğunu esas alan steganaliz örnek uygulaması

Bütün steganografik metotlar tespit edilebilen insan yapısı şeylerin girme ihtimalini en aza indirmek için en az miktarda deformasyon elde etmeye çalışırlar. Bununla birlikte, eğer üst görüntü başlangıçta JPEG formatında (ki sıklıkla durum böyledir) saklandıysa, uzaysal etki alanında gömülen mesaj JPEG sıkıştırması ile yaratılan karakteristik yapıyı rahatsız edecek, fakat silmeyecektir ve verilen bir görüntünün geçmişte JPEG olarak saklanıp saklanmadığı kolayca belirlenebilecektir. Kuşkusuz, JPEG sayısallaştırma Çizelgesinin bütün 8x8 bloklar içindeki DCT katsayılarının değerlerini dikkatle analiz ederek stego görüntüden tekrar elde etmek mümkündür. Bununla birlikte, üst görüntü, mesaj gömülmesinden sonra, her hangi bir sayısallaştırılmış katsayı bloğunun JPEG dekompresyonu ile belirli bir 8x8 piksel bloğunun üretilemeyeceğinin ispat edilmesinin mümkün olabileceği anlamında, JPEG formatı ile (büyük ihtimalle) bağdaşmayacaktır. Bu bulgu bloğun hafifçe tadil edildiği hususunda kuvvetli delil sağlamıştır. Kuşkusuz, güçlü bir JPEG sıkıştırması parmak izi taşıyan, ancak henüz her hangi bir JPEG sıkıştırılmış görüntüsü kadar uyumlu olmayan kayıpsız bir formatta saklanan bir görüntü bulmak şüphe uyandırıcıdır. Bu steganografinin delili olarak yorumlanabilir.

Her bloğun JPEG uyumluluğu kontrol edilerek, bir bit kadar kısa mesajlar potansiyel olarak tespit edilebilir. Steganalitik metot sadece LSB gömülüşü için değil, fiilen her hangi bir uzaysal steganografik veya filigran basma metodu için çalışacaktır. Hatta hangi 8x8 blokların JPEG sıkıştırması ile uyumlu olduğuna karar verilerek mesajın uzunluğu ve görüntü içindeki pozisyonunu kestirmeye teşebbüs edilebilir. Hatta görüntüyü analiz etmek ve üst görüntü veya blokları için muhtemel adayı ("en yakın" JPEG uyumlu görüntü/blok) kestirmek mümkündür. Bu şekilde, tadil edilmiş münferit pikseller tanımlanabilmektedir. *k'*inci bloğun *i*'nci DCT katsayısını $d_k(i)$, $1 \le i \le 64$, $k=1,\ldots,T$ ve *T*'yi görüntüdeki toplam blok sayısı olarak belirtelim. Her blokta, bütün 64 katsayı JPEG sayısallaştırıcı matrisi *Q*,, $D_k(i) = [d_k(i)/Q(i)]$ kullanılarak $D_k(i)$ tam sayılarına sayısallaştırılır. Burada $0 \le x \le 255$ için $[x]$= *integer_round (x)*, $x<0$ için $[x]=0$, ve $x>255$ için $[x]=255$'tir. DCT katsayısı $D_k(i)$ baskıdan kurtularak *B* bloğuna dönüşür:

$$B=[B_{raw}] \text{ , } B_{raw}=DCT^{-1}(QD), \quad \forall i \text{ için } QD=Q \otimes D \tag{3.11}$$

Denklem (3.11)'de, $\otimes$ matris elemanlarına göre çarpımı belirtmektedir. Ayrıca nütasyonu basitleştirmek için *k* blok endeksi de atılmıştır. JPEG sıkıştırması kayıplı olduğu için genel olarak B_{orig}'in *B*'ye eşit olmayabileceği belirtilmiştir. Eğer *B* bloğu 0 veya 255'te doygun piksele sahip değilse, L^2 normunda

$$\|B_{\text{raw}} - \text{B}\|^2 \le 16 \tag{3.12}$$

yazabiliriz, çünkü tam sayılara yuvarlama nedeni ile bütün $i=1,\ldots, 64$ için $|B_{raw}(i) B(i)| \le 1/2$.

Q sayısallaştırma matrisinin bilindiği varsayılsın. Steganalitik tekniği aşağıdaki soruyu esas almaktadır: Keyfi bir 8x8 piksel değerleri bloğu *B* verildiğinde, bu blok *Q* sayısallaştırma matrisi ile JPEG dekompresyonu işlemi vasıtasıyla mı ortaya çıkmıştır.

QD'=DCT(B) ifadesiyle, Parserval Eşitliği'ni kullanarak $\| B\text{-}B_{raw} \|^2 = \|DCT(B)\text{-}DCT(B_{raw})\|^2 = \|QD'\text{-}QD\|^2 \leq 16$ yazabiliriz. Diğer taraftan, $\|QD'\text{-}QD\|^2$ ifadesi için *QD(i)* yerine en yakın tam sayı katı *Q(i)*konularak daha düşük bir kestirimde bulunulabilmektedir:

$$16 \geq \|QD' - QD\|^2 \geq \sum_{i=1}^{64} | QD'(i) - Q(i)\, round\, [\frac{QD'(i)}{Q(i)}]| = S \qquad (3.13)$$

S niceliği sayısallaştırma matrisi *Q* bilindiği takdirde *B* bloğundan hesaplanabilir. Eğer *S* 16'dan büyükse, *B* görüntü bloğunun sayısallaştırma matrisi *Q* ile JPEG sıkıştırmasının uyumlu olmadığı sonucuna varılmaktadır. Tekrarlarsak bu sadece 0 veya 255'te doygunlaşan piksellere sahip olmayan bloklar için doğrudur. Kuşkusuz, (3.12) kestirimi doymuş pikselleri olan bloklar için geçerli olmayabilir, çünkü 0 ve 255'te yuvarlama ½'den çok daha büyük olabilir.

Eğer, doymamış pikselli bir *B* bloğu için *S* 16'dan küçükse, *B* bloğu JPEG uyumlu olabilir veya olmayabilir. $q_p(i)$, *p*=1,…., *Q(i)*'nin *QD(i)*'ye en yakın olan ve *QD(i)'*ye olan uzaklıklarına göre sıralanmış tam sayı katları olsun (en yakın katı $q_1(i)$). Verilen *B* bloğunun sayısallaştırma Çizelgesi Q ile JPEG sıkıştırmasıyla uyumlu olup olmadığına karar vermek için,

$$S = \sum_{i=1}^{64} |QD'(i) - q_{p(i)}\ (i)| \leq 16 \qquad (3.14)$$

için bütün { *p*(1),…., p(64)} indislerin 64-değişken setinin incelenmesi ve

$$B = [\ DCT^{-1}\ (QD)],\ \text{her}\ \ QD(i) = q_{p(i)} \qquad (3.15)$$

olduğunun doğrulanması gerekmektedir.

İfade (3.14)'i sağlayan 64-değişken seti { *p*(1),…., p(64)}sayısı her zaman sonludur ancak artan JPEG kalite faktörü ile hızla artar. 95'ten büyük kalite faktörleri için,

sayılaştırma faktörleri *Q(i)*'arın büyük kısmı 1 veya 2'dir ve bütün 64 indisin toplam birleşim sayısı ele alınamayacak kadar büyük olur.

Steganaliz algoritmasının tanımlanmanı:

1. Görüntüyü 8x8 bloklardan oluşan bir ızgara haline getirin ve eğer görüntü boyutları 8'in katları değilse son sıra ve sütunları atın.
2. Blokları bir liste halinde düzenleyin ve bütün doymuş blokları listeden çıkartın (eğer 0 veya 255 gri değerli en az bir pikseli varsa, blok doymuştur). Blokların toplam sayısını *T* olarak ifade edin.
3. Sayısallaştırma matrisi *Q'*yu bütün *T* bloklarından aşağıda tarif edildiği şekilde dışarı alın. Eğer *Q'*nun bütün elemanları birse, görüntü daha önce JPEG olarak saklanmamıştır ve steganalitik metodumuz uygulanmaz (bu algoritmadan çıkın). Eğer *Q* için birden çok akla yakın aday bulunuyorsa, 4–6 basamaklar bütün adaylar için tekrarlanmalı ve en yüksek JPEG uyumlu blok sayısını veren sonuç bu algoritmanın sonucu olarak kabul edilmelidir.
4. Her *B* bloğu için *S* niceliğini hesaplayın (Denklem 3.13). Eğer $S>16$ ise, *B* bloğu sayısallaştırma matrisi *Q* ile JPEG sıkıştırmasıyla uyumlu değildir. Eğer $S\leq16$ ise, her DCT katsayısı QD_i' için *Q(i)'*nin en yakın katlarını hesaplayın, bunları QD_i' ile olan mesafelerine göre düzenleyin ve $q_p(i),p=1,\ldots$. Olarak ifade edin. (3.14) eşitsizliğinin sağlandığı bu birleşim için ifade (3.14)'un sağlanıp sağlanmadığını kontrol edin. Eğer en az bir $\{p(1),\ldots, p(64)\}$ seti için (3.15) ifadesi sağlanmışsa, *B* bloğu JPEG uyumludur, aksi takdirde değildir.

Bütün *T* bloklarını gözden geçirdikten sonra, uyumlu olmayan her hangi bir JPEG bloğu bulunmamışsa, steganalitik metodumuzun gizli mesajların varlığına dair delil bulmadığı sonucuna ulaşılır. Diğer taraftan, eğer bazı JPEG uyumsuz bloklar varsa, gizli mesajın boyutunu kestirmeyi, mesajı taşıyan piksellerin yerini bulmayı ve hatta gizli mesaj gömme başlamadan önceki orijinal üst görüntüyü elde etmeye çalışabiliriz.

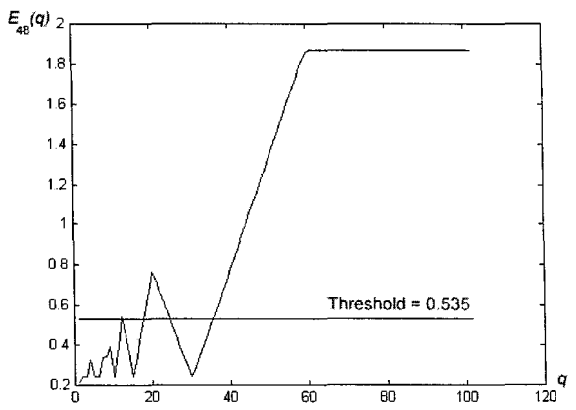
Eğer bütün bloklar JPEG uyumsuz olarak tanımlanmışsa veya görüntü daha önceden

JPEG olarak saklanmış görünmüyorsa, algoritmayı görüntünün değişik 8x8 bölümleri ile tekrarlanmalıdır (x ve y yönlerinde 0-7 piksel kaydırarak). Eğer üst görüntü mesaj gömülüşünden önce kırpılmışsa bu basamak gerekli olabilir.

Bu çalışmanın bir sonucu olarak, steganografik program kullanıcılarının steganografi için üst görüntüler olarak daha önce JPEG formatında saklanmış görüntüler kullanmaktan kaçınmaları gerekmektedir. JPEG sıkıştırma görüntü üzerinde benzersiz bir parmak izi bırakır ve bu parmak izi steganaliz için dikkat edilerek kullanılabilir. Başka görüntü mevcut değilse, JPEG parmak izinden kurtulmak için görüntüyü yeniden ölçeklendirmeye veya biraz daha küçük boyutlara indirmeye çalışılmalıdır. Filtreler kullanmak, zıtlık ayarlamak veya görüntüyü hafifçe döndürmekte JPEG parmak izini azaltmaya veya yok etmeye yardımcı olabilir.

Sayısallaştırma matrisinin türetilmesi:

Burada, Q sayısallaştırma matrisinin sıkıştırmadan kurtulmuş JPEG görüntüsünden nasıl elde edilebileceği gösterilmektedir. Bir önceki kısmın nütasyonunu aynen koruyarak, önce bütün doymamış 8x8 bloklardan $d_k(i)$, $0 \le i \le 64$, $k=1,\ldots,T$ DCT katsayılarını hesaplanır. Her i katsayısı için, $E_i(q)$ niceliğini $q=1,2,\ldots$'nin fonksiyonu olarak çizilir.



$$E_i(q) = \frac{1}{T}\sum_{k=1}^{T}\left| d_k(i) - q \times integer\_round\left(\frac{d_k(i)}{q}\right)\right|.$$

Şekil 3.16. 48-inci DCT katsayısı (13,11) için $E_{48}(q)$ hatası q sayısallaştırma basamağı (30'daki yerel minimumu ve 15,10,6,5,3 ve 2 bölenlerine dikkat edin. Son minimum 30 olup bu doğru Sayısallaştırmaya tekabül eder).

$E_i(q)$ niceliği görüntüdeki bütün i-nci DCT katsayılarının q sayısallaştırma basamağı ile uyumluluğunu ölçmektedir. Eğer incelenmekte olan görüntü önceden kuşkusuz JPEG olarak saklanmış olsaydı, doğru q sayısallaştırma değerinde ve q'nun bütün tam sayı bölenlerinde bir düşme (yerel minimum) gözleyecektik. Sayısallaştırma basamağının doğru değerinin E yerel minimumunun ortaya çıktığı yerdeki en büyük q değeri olması gerektiği sezgisel olarak açıktır. Bununla birlikte, daha önce tartışıldığı gibi, bu her durumda doğru olmayabilir.

Her şeyden önce, doğru değerden daha büyük q değerleri için bir takım "hayalet" yerel minimumların ortaya çıkması mümkündür. Bununla birlikte, bu minimumlar diğer "doğru" minimumlardan belirgin olarak daha büyüktür. Bir yerel minimumun en büyük muhtemel değerine bir eşik koyarak, sahte minimumları başarı ile "filtreden geçirebilir". Denemek de, eşiği DCT katsayılarına bağımlıdır (i endeksi üzerinde) ve μ_i ve σ_i $E_i(q)$ vektörünün bütün yerel minimumdur ortalama ve standart sapmaları olmak üzere, değerini $\mu_i+3\sigma_i$ olarak hesaplanmıştır. Eğer $E_i(q)$ vektörü, q=1 için olan hariç, her hangi bir yerel minimuma sahip değilse (bütün q>1 için $E_i(1)\leq E_i(q)$ dır), sayısallaştırma basamağının 1 veya 2 olduğuna karar vermek üzere $E_i(1)$ ve $E_i(2)$ arasındaki fark incelenmektedir. Bu durum için, çok iyi sonuçlar veren deneye dayalı bir kural geliştirilmiştir. Yapılan denemelerde dayanarak, eğer $E_i(1)<0.6x\ E_i(2)$, sayısallaştırma basamağının 1, aksi takdirde 2 olduğu sonucuna ulaşılmıştır.

Bu, her katsayı için sayısallaştırma basamaklarını bulma işlemi hala tamamen kusursuz değildir. Küçük görüntüler için, q sayısallaştırma basamağı için bütün $d_k(i)$ DCT katsayısı değerleri 2q'nun katları olacaktır. Bu durumda, doğru basamağın 2q mu, yoksa q mu olduğunu söylemek için aracımız yoktur. Bu olgunun olasılığı görüntü boyutu ile azalmaktadır. Bu probleme hitap etmek için, çok uzaktakiler için elde edilmiş Q sayısallaştırma matrisini standart JPEG sayısallaştırma Çizelgesine bölerek incelenir. Burada, özelleştirilmiş Çizelgelerin bile (yani dijital kameralardaki JPEG sıkıştırmada kullanılan Çizelgeler) standart sayısallaştırma Çizelgesinden büyük ölçüde sapmayacağı mantığını kullanmaktadır. Son olarak, şüpheli sayısallaştırma basamakları için adaylar tespit etmek ve bütün muhtemel birleşim

için steganaliz yapmak her zaman mümkündür.

3.6. Evrensel Kör Steganaliz

Her hangi bir uzaysal steganografik metoda uygulanabilen JPEG uyumluluk steganalizi hariç, önceden önerilmiş bütün metotlar özel bir gömme algoritması veya varyasyonları için ısmarlama olarak yapılmıştır. Evrensel kör steganalizi orijinal ve stego görüntüler üzerinde eğitimden sonra gömme etki alanı dikkate alınmaksızın her hangi bir steganografik metodun tespiti için ayarlanabilir olması anlamında tespit ötesi bir metottur. Marifet "ayırt edici" yeteneklere sahip uygun bir hassas istatistik nicelikleri kümesi (bir özellik vektörü) bulmaktır. Böylece yapay sinir ağları, kümelenme algoritmaları ve diğer yapay zekâ araçları doğru eşikleri bulmak ve toplanan deneysel verilerden tespit modelini inşa etmek için kullanılabilir.

Farid, stego görüntünün dalgacık parçalanmasından türetilmiş yüksek seviyeli hassas istatistikler kümesi önermektedir [16]. Sonra, özellik vektörlerini Fisher Doğrusal Ayırma analizini kullanarak, biri stego görüntülere, diğeri orijinal görüntülere karşılık gelen iki doğrusal alt uzaya bölmektedir. Karar eşiği sahte pozitifler gözden kaçırılmış tespitlerle takas etmek üzere ayarlanabilir. Bu yaklaşımın oldukça keyfi "geçici" tercihler içerdiği gerçeği göz önünde tutulursa, metodunun nasıl bu kadar iyi çalıştığı dikkate değer.

Farid'in yaklaşımı stego görüntünün, $V_i(x,y)$, $H_i(x,y)$, $D_i(x,y)$ i ölçeğinde düşey, yatay ve diyagonal alt bantları ifade etmek üzere, n-inci seviye dalgacık parçalanması ile başlamaktadır. Sonra, bütün üç alt bant için bütün i=1,….,n-1 seviyeleri için ilk dört momenti hesaplamaktadır. Bu toplam 12(n-1) istatistiksel nicelik verir. Sonra, en uygun doğrusal tahminci kullanarak dalgacık katsayısının fiili değeri ile en uygun doğrusal tahmin arasındaki tahmin hatası için uzaysal, yöneltme ve ölçek komşularından (toplam 7 komşu) aynı istatistiksel momentleri toplamaktadır. Örneğin, düşey dalgacık katsayıları aşağıdaki doğrusal tahminciyi kullanarak tahmin edilmektedir:

$V_i\,(x,y) = w_1\,V_i\,(x-1\,,\,y) + w_2\,V_i\,(x+1\,,\,y) + w_3\,V_i\,(x\,,\,y-1) + w_4\,V_i\,(x\,,\,y+1) + w_5\,V_{i+1}\,(x/2\,,\,y/2) + w_6\,D_i\,(x\,,\,y) + w_7\,D_{i+1}\,(x/2\,,\,y/2)$.

Karesi alınmış tahmin hatasını en aza indiren ağırlıklar standart en küçük kareler metodu kullanılarak hesaplanır. $V_i(x,y)$ için doğrusal tahmin $\tilde{v}_i(x,y)$ şeklinde ifade edilerek, tahminin log hatası $E_v(i)=\log_2(V_i)-\log_2(\tilde{v}_i)$ olarak tanımlanır. Farid $E_v(i)$'nin ilk dört momentini özellik vektörünün başka bir parçası olarak hesaplamaktadır. İşlemin bütünü bütün n-1 ölçekler için yatay ve diyagonal alt bantlarla tekrarlanmaktadır. Böylece, özellik vektörünün nihai uzunluğu $12(n-1)+4x3(n-1)=24(n-1)$ olmuştur. Özellik vektörü orijinal görüntülerin ve sabit boyutlu mesaj gömülü stego görüntülerin geniş bir veri bankası için hesaplanmıştır. Özellik vektörlerini bir eşiğin ayırdığı iki kümede sınıflamak için Farid Fisher Doğrusal Ayırma (FLD) analizini kullanmaktadır [87]. Bu yaklaşımın avantajı FLD analizinin tek boyutta hızlı ve basit skelar eşik oluşturmayı sağlamasıdır.

Farid tarafından bildirilen sonuçlar [87], yaklaşımının çok etkileyici sonuçlar veren tutarlı bir yöntem olduğunu göstermektedir. Farid J-Steg, EZ Stego, ve OutGuess'in her iki sürümünü kullanarak küçük bir 256x256 piksel gri ölçek görüntü gömdüğü 1400x1000 yüksek kaliteli görüntülerin veri tabanını kullanmıştır. En iyi sonuçlar 1,8% sahte pozitiflerde 97,8% tespit güvenirliliği ile J-Steg için elde edilmiştir. EZ Stego 13,2% sahte pozitif ile 86,6% güvenirlilikle tespit edilmiştir. OutGuess için aynı sonuçlar 80,4% ve 19,4% (sürüm 0,1 için), ve sürüm 0,2 için 77,7% ve 23,8% olmuştur. Bu kısmı RS steganalizi gibi, özellikle özel bir steganografik metoda hedeflenmiş metotların her hangi bir evrensel kör steganalitik metottan muhtemelen daha doğru ve güvenilir sonuçlar vereceğini belirterek bitirelim. Mamafih, evrensel kör yaklaşımlar esneklikleri ve yeni veya hiç bilinmeyen steganalitik metotlara kolayca ayarlanma yetenekleri nedeniyle çok önemlidirler.

3.7. Benzersiz Parmak İzleri

Bazı steganografik teknikler veya özel uygulamaları tamamen benzersiz ve kolayca tespit edilebilen insan yapısı eserler yaratmaktadır. Örneğin, Los Alamos tasarımının

demo sürümü stego görüntünün son satırını inceleyerek hızla kırılabilmektedir, çünkü bu sıra bir yan kanal olarak hizmet etmekte ve gömme için kullanılan renk çiftleri ile bilgi içermektedir [1]. GIFshuffle rastgele düzenlenmiş paletlerle görüntüler üretmektedir ki, bunlar da şüphe çekici ve kolayca kontrol edilebilen ürünlerdir [1]. S-Tools görüntü paletini önceden işlemekte ve gömme için takas edilen çok yakın renk kümeleri yaratmaktadır. Görüntü paletinin basit bir analizi gizli mesajların varlığına işaret edebilir. Anderson'un bir makalesinde daha çok örnek bulunabilir [1].

3.8. RQP Steganaliz

RQP yöntemi Fridrich, Du R., Meng L., tarafından geliştirmiştir [39]. Bu metot LSB gizlemesi tarafından yaratılan yakın renk çiftlerini analiz etmeye yöneliktir. Öncelikle seçilen resim için yakın renk çiftlerinin tüm renk çiftlerine oranı hesaplanır. Daha sonra bu resim içerisine bir test mesajı gizlenerek oran yeniden hesaplanır. Bu iki oran arasındaki fark büyük ise resminin içinde gizlenmiş bilgi yok demektir. Bu iki oranın birbirine yakın olması resmin içinde gizlenmiş bilgi olduğunu göstermektedir.

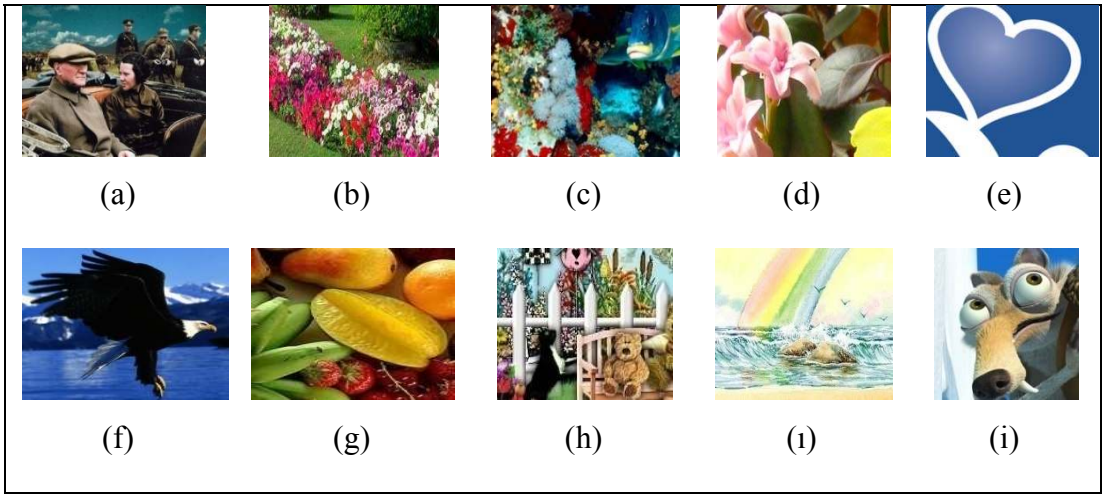
RQP, örtü verisindeki yakın renk çiftlerinin sayısı, piksellerin sayısının %30'undan küçük olduğu sürece gayet iyi sonuçlar vermektedir. Eğer görüntüdeki yakın renk çiftlerinin sayısı piksellerin sayısının %50'sini geçerse, verilen sonuçlar giderek güvensiz olmaktadır. RQP'ın başka dezavantajı, gri seviyeli görüntülerde uygulanmamasıdır.

3.8.1. RQP Steganaliz örnek uygulaması

Program BMP formatındaki 24 bit renkli resimler üzerinde çalışmaktadır. Öncelikle seçilen resim için yakın renk çiftlerinin tüm renk çiftlerine oranı (O_1) hesaplanmaktadır. Daha sonra bu resim içerisine bir test mesajı gizlenerek oran (O_2) yeniden hesaplanır.

Bu iki oran arasındaki farkın büyük olması resminin içinde gizlenmiş bilgi olmadığını göstermektedir. Bu iki oranın birbirine yakın olması ise resmin içinde gizlenmiş bilgi olduğunu göstermektedir. Fakat bu büyüklük ve küçüklük göreceli bir kavramdır. Aradaki farkın nasıl yorumlanması gerektiğini tam olarak belirleyebilmek için birçok resim üzerinde ölçümler yapılmıştır.

Programın çalışmasını incelemek amacıyla örnek olarak 10 adet resim seçilmiş ve Şekil 3.17'de gösterilmiştir. Öncelikle içinde bilgi gizli olmayan resimlere RQP Steganaliz uygulanmış ve elde edilen sonuçlar Çizelge 3.3'de verilmiştir. Daha sonra aynı resimlerin içerisine bir metin gizlenmiştir ve tekrar RQP Steganaliz uygulanmıştır



Şekil 3.17. RQP Steganaliz için kullanılan örnek resimler [22].
a) Atatürk.BMP 400x300 piksel.
b) Bahçe.BMP 335x192 piksel.
c) Balık.BMP 379x253 piksel.
d) çiçek.BMP 312x223 piksel.
e) kalp.BMP 313x292 piksel.
f) kartal.BMP 269x249 piksel.
g) meyve.BMP 217x238 piksel.
h) oyuncak.BMP 240x192 piksel.
ı) resim.BMP 336x240 piksel.
i) scrat.BMP 292x308 piksel

Resmin içine bilgi gizleme işlemi tarafımızdan geliştirilen Stego_LSB isimli program tarafından yapılmıştır [9]. Bunun sonucunda elde edilen değerler ise Çizelge

3.4'de gösterilmiştir. İçinde bilgi gizli olan ve olmayan resimler için O_1 ve O_2 değerleri arasındaki farklar incelenmiştir Programın sahte kodu aşağıda verilmiştir.

Adım 1: Resmi seç.
Adım 2: Yakın renk çiftlerinin sayısını hesapla (renk çiftleri arasındaki fark 3'ten küçük olanlar yakın renk çifti olarak seçilmiştir.)
Adım 3: Yakın renk çiftlerinin tüm renk çiftlerine oranını hesapla ve O_1 olarak belirle.
Adım 4: Seçilen resmin içine bir test mesajı gizle ve oranı tekrar hesaplayıp O_2 olarak belirle.
Adım 5: O_1 ile O_2 arasındaki farkı hesapla.

Burada renk çiftlerinin arasındaki yakınlığın ne kadar olacağı da önemlidir. Bu çalışmada renk çiftleri arasındaki yakınlık 3 olarak alınmıştır. Kırmızı, yeşil ve mavi renk kanalları için ayrı ayrı olmak üzere pikselleri arasındaki renk farkları değerlendirilmiştir.

Çizelge 3.3. İçine bilgi gizlenmemiş resimlere uygulanan RQP steganaliz sonuçlar

	O1	*O2*	*Fark*
Atatürk.BMP	0,30413	0,28312	0,02101
Bahçe.BMP	0,10332	0,09098	0,01235
Balık.BMP	0,23569	0,22222	0,01347
çiçek.BMP	0,35365	0,32045	0,03320
kalp.BMP	0,91485	0,91125	0,00360
kartal.BMP	0,68488	0,65845	0,02643
meyve.BMP	0,31941	0,28589	0,03352
oyuncak.BMP	0,12483	0,11404	0,01079
resim.BMP	0,39259	0,37610	0,01649
scrat.BMP	0,40881	0,38609	0,02272

Çizelge 3.4. İçinde bilgi gizli olan resimlere uygulanan RQP steganaliz sonuçları

	O2	*O2*	*Fark*
Atatürk.BMP	0,28615	0,28312	0,00302
Bahçe.BMP	0,09311	0,09098	0,00213
Balık.BMP	0,22408	0,22222	0,00186
çiçek.BMP	0,32547	0,32045	0,00502
kalp.BMP	0,91182	0,91125	0,00057
kartal.BMP	0,66287	0,65845	0,00442
meyve.BMP	0,29172	0,28589	0,00583
oyuncak.BMP	0,11612	0,11404	0,00208
resim.BMP	0,37729	0,37610	0,00119
scrat.BMP	0,38988	0,38609	0,00379

Çizelge 3.3 ve Çizelge 3.4'taki [22] değerlerinden de görüleceği gibi içinde bilgi gizli olmayan resim dosyalarına uygulanan RQP steganaliz sonucunda fark değerlerinin yüzde seviyesinde olduğu görülmektedir. İçinde bilgi gizli olan dosyalarda ise bu fark binde seviyesine düşmektedir. Bu nedenle programın çalışması sonucunda elde edilen değerler binde seviyesinde ise resim içinde bilgi gizlenmiştir denilebilir.

Yapılan denemeler sonucunda programın, görüntüdeki yakın renk çiftlerinin sayısının piksellerin sayısının %50'sini geçtiği durumlarda da doğru sonuçlar verdiği görülmüştür.

RQP yöntemi Fridrich ve arkadaşları tarafından geliştirmiştir [39]. Bu metot LSB gizlemesi tarafından yaratılan yakın renk çiftlerini analiz etmeye yöneliktir. Öncelikle seçilen resim için yakın renk çiftlerinin tüm renk çiftlerine oranı hesaplanır. Daha sonra bu resim içerisine bir test mesajı gizlenerek oran yeniden hesaplanır. Bu iki oran arasındaki fark büyük ise resminin içinde gizlenmiş bilgi yok demektir. Bu iki oranın birbirine yakın olması resmin içinde gizlenmiş bilgi olduğunu göstermektedir.

RQP, örtü verisindeki yakın renk çiftlerinin sayısı, piksellerin sayısının %30'undan küçük olduğu sürece gayet iyi sonuçlar vermektedir. RQP, gizli mesajın büyüklüğünün hakkında sadece iyi bir tahmin sağlamaktadır. Eğer görüntüdeki yakın renk çiftlerinin sayısı piksellerin sayısının %50'sini geçerse, verilen sonuçlar giderek güvensiz olmaktadır. RQP'ın başka dezavantajı, gri seviyeli görüntülerde uygulanmamasıdır.

Çizelge 3.5. Steganaliz Tekniklerinin Karşılaştırılması

Steganalitik metodu	Tanımlama	Hedeflenen Steganografik teknikleri
RS steganalizi	Yüksek dereceli istatistiksel analizler ise histogram analizinden fazla olarak değişimin gerçekleştiği piksellerin konumlarına yönelik istatistiksel analizleri de içermektedir.	Çeşitli LSB modifikasyon teknikleri
Ki-kare testi	Ki-kare testi, sabit değer çiftleri seti mesaj bitlerini gömmek için birbirinin karşılaştırmaktadırlar.	Piksel gri seviye, renk Ya da DCT katsayı değer çiftlerinin değiş tokuşuna dayalı Steganografi yöntemi
Palet Denetimi (Palette checking)	Palet sırlamasında özellik, sistematik değişikliğin açık bir işaretidir	Palet şekillerinde Steganografi
RQP metodu	Metot gömme işlemlerinden kaynaklanan yakın-renk çiftlerinin artan sayısına dayalı yöntem.	Gerçek-renk şekillerinde gömerken LSB
JPEG Uyumluluğunu Esas Alan Steganaliz	Algoritması bilinen steganografi uygulamalarıyla oluşturulan stego resimlerdeki mesajı ortaya çıkarmaya yönelik yöntemlerle JPEG gibi resim türlerine özel steganaliz çalışmalarını içermektedir.	Boşluk-alan steganografi kullanma şekilleri önce, JPEG biçiminde depoladı
Histogram analizi	Histogram analizleri resme veri gömme sonucu değişen piksel değeri çiftlerinin istatistiksel analizlerini esas almaktadır.	QIM veya diğer nicelendirme dizin gömme metotları
Evrensel Kör Steganalizi	Evrensel tespit sistemleri ise resme ait bazı özelliklerin çıkarımıyla orijinal resimlerle stego resimler arasındaki ayırımı yapmaya yönelik yöntemleri içermektedir.	Çeşitli steganografik teknikleri

4. MEVCUT STEGANALÎZ YAZILIMLARI

Steganografinin varlığını tespit edebilen ve steganaliz yazılımı olarak adlandırılan bir takım araçlar bulunmaktadır. Bunların bazıları açık kaynaklı olup diğerleri bir hayli pahalıdır. Bu çalışma serbest olarak elde edilebilen araçların test edilmesi sınırlıdır, bu yüzden sadece kamuya açık tariflere dayanan lisanslı seçenekleri ele almaktadır.

Gizli bilgi tespit edildikten sonra, gömülü mesaj çıkarılmaya çalışılmaktadır. Bazı durumlarda, bu tanımlanan steganografi yazılımının çalıştırılması anlamına gelmektedir. Bazen, mesajı çekip çıkartmak için bir anahtar gerekmektedir. Bu sistemler üzerinde kaba kuvvet veya sözlük saldırısı uygulanabilir. Steganografi yazılımına karşı sözlük saldırılarını destekleyen bir yazılım örneğinden de ayrıca bahsedilmiştir. Çizelge 4.1 bu bölümde test edilen steganaliz yazılımlarının genel bir açıklamasını vermektedir.

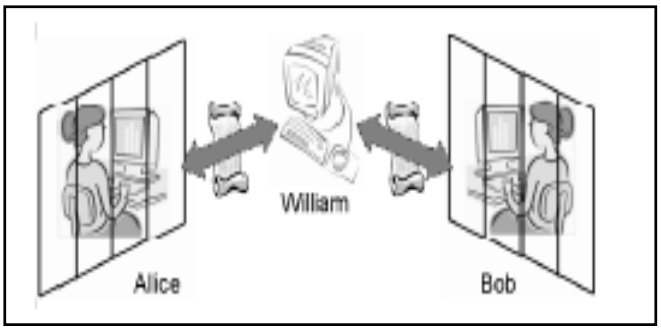
Çizelge 4.1. Steganaliz yazılımları

Adı	Kısmı	Lisans Durumu
StegSpy	4.1	Açık kaynak
Stegdetect	4.2	Açık kaynak
Stegbreak	4.3	Açık kaynak
Stego Suite	4.4	Lisanslı
StegAnalyzer	4.5	Lisanslı

Her aracın kullanılışı önce tarif edilmiştir. Daha sonra araçlar özellikle sınırlamalarından söz edilerek incelenmiştir. Bunlardan bazılarına aşağıda değinilmiştir:

Gizli bilginin seçilemez kılınışı: Gizli bilginin seçilemez kılınışına ait yazılım ele alınmamıştır. Genellikle gömme metodu stego mesajındaki küçük değişikliklere bile dayanıklı değildir. Görüntüler için, bu bilgi azaltıcı sıkıştırma, yeniden boyutlandırma, vb. olabilir. Gizli bilginin seçilemez kılınışını engellemek, filigran basma planının hedefidir. Örneğin, Digital Rights Management (DRM) buna

bağlıdır. Faal bekçi olayında (The Prisonners Problem), bekçi (warden) William potansiyel gizli bilgiyi yok etmek için metotlar uygulayabilirdi [62].



Şekil 4.1. Faal bekçi olayı (The Prisonners Problem)

4.1. Stegspy

StegSpy yazılımı dosyada veri saklı olup olmadığını ve saklıysa hangi yazılımla saklandığını tespit edebilmektedir. Evrensel tespit yapamamakta sadece, JPHideandSeek, Invisible Secrets gibi birkaç steganografik yazılımla oluşturulan stego dosyaları tespit etmektedir. Yazılım ücretsiz olarak indirilebilmektedir.

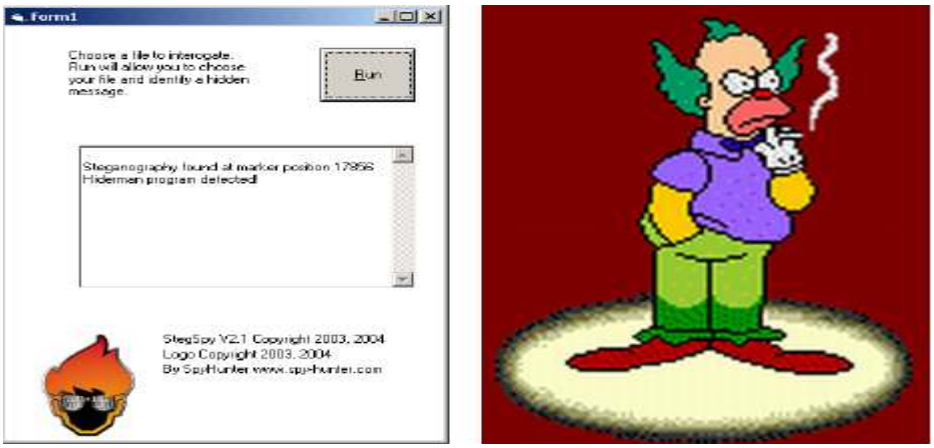
Çizelge 4.2. StegSpy V2.1

Adı	StegSpy V2.1
Lisans durumu	Parasız
Tespit edilen yazılım	Hiderman, JPHideandSeek, Masker, JPegX ve Invisible Secrets
URL	www.spy-hunter.com/stegspydownload.htm

StegSpy V2.1 Michael T. Raggo tarafından geliştirilmiş parasız elde edilebilen steganaliz yazılımıdır. Hiderman, JPHideandSeek, Masker, JPegX ve Invisible Secrets tespit ettiğini ileri sürmektedir [63].

Müellifi StegSpy'ı InfoSec 2004, BlackHat 2004 ve DefCon 2004'te sunmuştur [64].

StegSpy'ın mevcut sürümü v2.1 Visual Basic dilinde yazılmıştır. Kullanıcının incelenecek bir dosyayı manüel olarak seçmesine izin veren grafik ara yüzü bulunmaktadır. Şekil 4.2(a)'da bir ekran resmi görülmektedir. Şekil 4.2(b)'deki resim M.T. Raggo tarafından BlackHat 2004'teki steganografi ve steganaliz sunumundan alınmıştır.



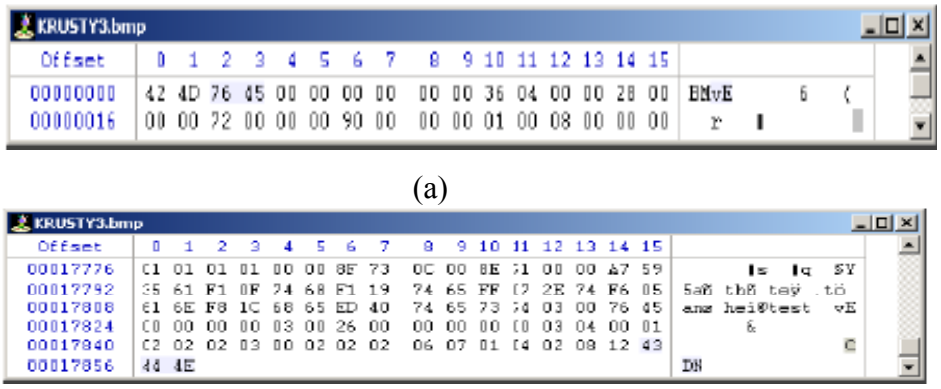
(a) (b)

Şekil 4.2. StegSpy kullanımı.
a) StegSpy v2.1 ekran resmi
b) Palyaço Krusty'nin stego görüntüsü (KRUSTY3.BMP)

Şekil 4.2(a)'da StegSpy Hiderman ile pozisyon 17856'da başlayan saklanmış bilgi bulunduğunu göstermektedir. Stego dosyası BMP görüntü verisidir. BMP sayfa başlığı dosyanın boyutunu belirten bir alan içermektedir [7]. Şekil 4.3 görüntünün üst ve alt kısımlarını bir onaltılık sistemin (hexadecimal) göstericide göstermektedir. Şekil 4.3 ekli veriler bulunduğunu açıkça belirtmektedir.

Şekil 4.3(a)'deki sayfa başlığı dosya boyutunun 17782 (0x4576) bayt olduğunu belirtmektedir. Bununla birlikte, bu göreli konumu aşan veriler bulunmaktadır.

Şekil 4.2(b)'de ayrıca "CDN" dizgisi de görülmektedir. Hiderman [9] kullanılırken bu dizgi her zaman mevcuttur. Öyleyse CDN Hiderman'ın imzasıdır ve mesaj gömme için kullanılan steganografi yazılımını tespit etmek için StegSpy tarafından kullanılmaktadır.



(a)

(b)

Şekil 4.3. Bir onaltılık sistemin (hexa) göstericide KRUSTY3.bmp'nin görünüşü.
a) BMP görüntü dosyasının sayfa başlığının başlangıcı (KRUSTY3.BMP).
b) (KRUSTY3.BMP)'nin sonu.

Müellife göre [9], StegSpy imzaya dayanan steganaliz yapmaktadır. StegSpy'ın nasıl çalıştığı hakkında çok şey bilinmemektedir, ancak dosya aykırılıklarını ve imzaları tespit eden stego: File Signatures kısımlarında tarif edilen benzer bir yaklaşımı izlediği söylenebilir. Bununla birlikte, "CDN" imzasını "000" olarak değiştirince, StegSpy KRUSTY3.BMP içindeki steganografiyi tespit etmeyi başaramamaktadır. Böyle olunca, StegSpy sadece dosya imzasına güveniyor ve tespit için dosya aykırılıklarını göz ardı ediyor gözükmektedir. Bilinen bir imza tespit edildikten sonra, gömülü verilerin pozisyonunu belirlemek için bazı çalışmalar yapmaktadır, örneğin sayfa başlığı bilgisine dayanarak görüntünün sonunu tespit etmek.

Grafik kullanıcı ara yüzü steganografi için dosyaların böyle seçilmesine izin vermemektedir. Bu yüzden çok sayıda muhtemel dosya arasından steganografi için arama yapmak pek kullanışlı olmamaktadır. Yukarıdaki gözlemlere dayanarak, StegSpy'ın hukuksal kullanım değeri düşüktür. Ancak içerdiği bilgi, örneğin imzalar, bir hayli yararlıdır ve bu imzalar kullanılarak daha hukuk dostu bir araç yaratmak için bir çalışma yapılabilir.

4.2. Stegdetect

Outguess 0.1**,** Invisible Secrets gibi yazılımlarla içerisine veri saklanmış JPEG resimlerini tespit edebilen ücretsiz bir steganaliz yazılımıdır [38].

Çizelge 4.3. Stegdetect 0.6

Adı	Stegdetect 0.6
Müellifi	Niels Provos
Lisans durumu	Açık kaynak
Tespit edilen yazılım	jsteg, jphide, Invisible Secrets, Outguess 0.13b, F5, appendX ve camouflage.
URL	www.outguess.org/detection.php

Stegdetect Niels Provos tarafından geliştirilen açık kaynaklı steganaliz yazılımıdır. jsteg, jphide (unix ve Windows), Invisible Secrets, Outguess 01.3b, F5 (sayfa başlık analizi), AppendX ve camouflage ile gömülen bir mesajın varlığını tespit edebilir [8].

Çizelge 4.4 Stegdetect'in Invisible Secrets kullanılarak gömülen bir mesajın bulunduğu bir görüntü üzerinde çalıştırılmasının çıktısını göstermektedir.

Stegdetect sonuçların kesinliğini *'ler- ne kadar çok olursa o kadar kesin belirtmektedir.

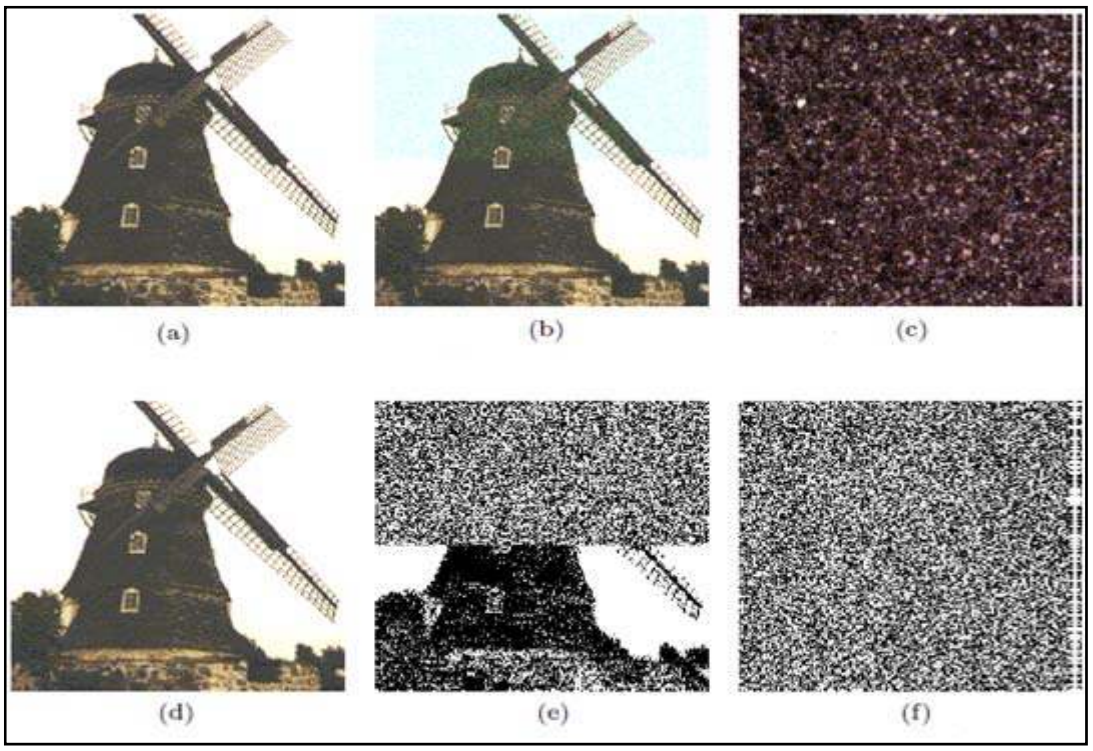
Çizelge 4.4. Stegdetect'in çalıştırılması

```
#./stegdetect – 0.6/stegdetect img1.jpg
img1.jpg: Invisible [7771] (***)
```

Stegdetect'in nasıl çalıştığı şöyledir: Steganografi algoritmaları, fark edilebilir değişiklikler arz etmeyecek olan taşıyıcı alanları içine mesajlar gizlemeye çalışır, hatırlayalım; C = p + t. Ancak, taşıyıcıda istatistikî değişiklikler olduğu keşfedilmiştir. Yani, C ve C' arasında istatistikî değişiklikler mevcuttur. Sözü edilmekte olan bu değişiklikler, steganografi algoritmalarının kırılması için kullanılabilir durumdadır. Örneğin, görüntüler için bir norm oluşturmak sureti ile yani, muhtemel taşıyıcılar, stego görüntüleri bu normdan çıkarım yapacaklardır. Bazı testler veri formatlarından bağımsızdır ve sadece gereğinden fazla olan verilerin dağılımını (entropisini) ölçer. Stego görüntülerinin daha yüksek bir dağınım (entropi)

faktörlerinin olması beklenir [68]. İstatistikî stego analizi için çok çeşitli yöntemler mevcuttur. Bundan sonrakiler, literatürdeki çok çeşitli yöntemlerin bir sunumunu takip etmektedir. “Bir uzamsal alan örneği, görüntülerin en az belirgin olan bit (LSB) plakası içine bir veri gizlemektir. Bu, görüntüye ait LSB’nin (en az belirgin olan bit), rastgele bir ses olarak düşünülebileceği varsayıma dayanmak sureti ile düzenlenmiştir. Gerçek gizleme çok sayıda yaklaşım gerektirmektedir: ardışık değişiklikler, bir sahte rastlantısal jeneratör kullanılmak sureti ile rastgele bir tur, değer eşitliği fonksiyonları, vesaire”. Kapsamında tanımlanmakta olan LSB (en az belirgin olan bit) barındıran uzamsal alan çok sayıda steganografi yazılımı kullanılması sureti ile elde edilmiştir, örneğin; EzStego. LSB’nin (en az belirgin olan bit), rastgele bir ses olması varsayımı doğru değildir. Bu nedenle, stego görüntüsünün ve kapak görüntüsünün istatistikî özellikleri birbirlerinden farklıdır ve [72] ile de kanıtlandığı üzere, bunlar tespit edilebilmektedir [70]. JPEG görüntülerini taşıyıcılar olarak kullanan steganografi sistemlerine atak yapmaktadır. JPEG algoritmaları, JPEG görüntüleri üzerinde ayırt edilebilir parmak izleri bırakmaktadır ve [70]’den alınan stego analiz yöntemi bunları “kırılgan filigranlar” olarak kullanmaktadır. Bu filigranların yok edilmiş olması halinde, steganografinin mevcudiyeti varsayılır.

Bir rastlantısal pikselin LSB’sinin (en az belirgin olan bit) modifiye edilmesi kadar küçük saklı değişiklikleri tespit etmek için. Bu yöntem, uzamsal alan saklama karşısında iyi çalışmaktadır ancak ayrık kosinüs dönüştürme (DCT) katsayılarının kullanımı ile steganografi algoritmaları karşısında verimli değildir. Yukarıda ifade edilmekte olan görsel atağa ek olarak [72] Ki-kare atağını temsil etmektedir. Bunlar Değerler Çifti (DÇ) kavramını arz ederler. EzStego ve LSB (en az belirgin olan bit) gizlenimini dikkate alalım, bu süreç sadece LSB’lerinde (en az belirgin olan bit) farklılıklar gösteren değer çiftlerini ortaya çıkarır. Bir kapak görüntüsü için, renk histogramı düzensiz niteliklidir. Bir mesajın gizlenmesinden sonra, bütün LSB’lerde (en az belirgin olan bit) eşit olarak dağıtımı yapılmış olan bitler ile her bir Değer Çifti (DÇ) meydana gelişi eşit hale gelir.



Şekil 4.4. Görsel atak adımları [72].
a) Taşıyıcı-imaj
b) Stego-imaj %50 veri gömülmüş
c) Taşıyıcı-imaj2
d) a'nın Filtre durumu
e) b'nın Filtre durumu
f) c'nın Filtre durumu

Taşıyıcıya ait bütün LSB'nin (en az belirgin olan bit) kullanılmamış olması halinde, mesajın sonunda, istatistiklerde bir değişiklik gözlemlenir. Değer Çiftleri (DÇ) hep çiftler halinde olabilirler ve bunlar mesajın saklanması sırasında birbirleri ile değiştirilirler. [72]'de belirtilmekte olan ile aynı istatistikî atak, [10]'nin kullanılması sureti ile JPEG görüntülerine ait ayrık kosinüs dönüştürme (DCT) üzerinde yapılır. Bu çalışma Provos tarafından daha detaylı bir istatistiksel analiz tarifi [10]'de bulunabilir. Stegdetect akademik çevrelerden çıkmış bir sonuçtur. Daha yeni teorik steganografik algoritmalar önerilmiştir[1], ancak bunları destekleyecek kamuoyu tarafından bilinen steganaliz yazılımı bulunmamaktadır[2].

[1] Örn[5] Outguess 0.2'yi daha yüksek dereceli istatiksel saldırılarla tespit etmektedir.
[2] Gene, lisanslı steganaliz yazılımı performansı incelenmemiştir.

Stegdetect 0.6 aynı zamanda her hangi bir JPEG tabanlı steganografi sistemini tespit etmek için doğrusal ayrıştırma analizini desteklemektedir [8]. Taşıyıcı görüntüler ve stego görüntüler eğitim seti olarak kullanılmaktadır. Test setini esas alarak bir doğrusal karar fonksiyonu otomatik olarak yaratılmakta ve yeni görüntüleri stego görüntüler veya gizlenmiş görüntüler olarak tasnif etmek için kullanılmaktadır. Bu işlevsellik incelenmemiştir.

4.3. Stegbreak

Çizelge 4.5. Stegbreak

Adı	Stegbreak
Müellifi	Niels Provos
Lisans durumu	Açık kaynak
Tespit edilen yazılım	JSteg-Shell, JPHide ve Outguess 0.13b
URL	www.outguess.org/detection.php

Stegbreak Stegdetect gibi aynı müellif, Niels Provos, tarafından geliştirilmiştir. Bununla birlikte, steganografinin varlığını tespit etmek için değil, ama mesaj çıkarımı için bir yazılımdır. Jsteg-Shell, JPHide ve Outguess 0.13b'ye karşı sözlük saldırısını dener [8]. Stegbreak'in başarısı kuşkusuz şifrenin ve sözlüğün kalitesi ile yakından ilgilidir. Sözlükteki kelimelerin sırasını değiştirme kuralları da başarı ile yakından ilgilidir[3].

Hukuksal bir bakış açısından, permütasyonların kullanılan şifre olabileceği şekilde, araştırma şifrelerin ipuçlarını sunabilir. Ad, doğum günü, evcil bir hayvanın adı, vb. gibi bu ipuçları Stegbreak tarafından kullanılan sözlüğe eklenebilir.

Stegbreak çıkarılmış bit dizgisinin sadece ses değil, ancak gömülü bir mesaj olduğunu doğrulayacak bir metoda gerek duyar.

[3] Stegbreak'in daha yeni sürümleri sözlük permütasyonları ve sözlükler için kurallar içermemektedir. Windows için Stegdetect 0.4 kullanılabilecek bazı kurallarla gelmiştir.

Saklı bilginin varlığının bilinmesinden ya da saklı bilginin varlığından şüphe edilmesinden sonra, gizlenmiş olan mesajı çıkarmaya çalışan yöntemler bir sonraki adım olacaktır. Bu keşfedilmiş olan steganografi algoritmasının ya da aracının kullanımı ile mümkün olabilecektir. Basit steganografi ile bu göreceli olarak kolaydır. Bir stego anahtarının mevcudiyeti halinde, bu anahtar bilginin çıkarılmasında başarının elde edilmesi açısından gereklidir. Bütün steganografi kullanıcılarının mesajlarını şifreleyebilme yetenekleri olmamasına rağmen, bu tahmin edilebilir. Şifrelenmiş olan dosyaların başlıklarının silinmesi, sesten ayırt edilemez olan bir gizlenmiş mesaj ile sonuçlanır. PGP Stealth 4, PGP şifreli bir mesajın bütün başlıklarını ayıklayan bir araçtır. Bundan sonra, kaba kuvvet araştırmasının karmaşıklığı, çok daha büyüktür, çünkü her bir stego anahtarı için Ks, bütün şifreleme anahtarlarının denenmesi gerekmektedir. Sadece gizlenmiş olan mesajın başarılı bir şifre analizi mevcut durumda denenmekte olan KS'nin doğru olan olup olmadığını, ya da öyle farz edilip edilmediğini gösterir. Ancak, [69] O (lKSl) olan bir teknik göstermektedir; bundan dolayı, stego anahtar araması, kriptografik anahtar büyüklüğünden bağımsız durumdadır. Ancak, [69] aynı zamanda da sunulmakta olan stego anahtarı araştırmasına karşı tedbirleri de temin etmektedir; sözü edilmekte olan bu tedbirlerin de steganografi algoritması tarafından adapte edilmesi mümkündür.

Sunulduğu gibi, çıkarılmış bit dizgisindeki dosya başlıklarını tanımlayarak yapılır.

4.4. Stego Suite

Çizelge 4.6. Stego Suite

Adı	Stego Suite
Müellifi	Wetstone Technologies
Lisans durumu	Lisanslı
Tespit edilen yazılım	Bilinmiyor
URL	www.wetstonetech.com\

Son on yıl içinde, görüntü, video ve ses verilerini sayısal olarak işleme teknolojileri çok büyük ölçüde gelişerek bilgilerin ikili veri dosyaları içinde hızla saklanabilmesine yol açmıştır. Sayısız web sitesi indirmek üzere "stego" programları teklif etmektedir. Sanayi casusluğu, ticaret sırrı hırsızlığı, siber silah değiş tokuşu ve kriminal koordinasyon potansiyeli sınırsızdır.

Birçok devlet ve ticari kurum büyük ölçüde kaygılı olup her şekliyle sayısal steganografi kullanımını tespit etmek için araçlar aramaktadırlar. Stego Suite böyle bir araç olup hedef dosyaya karşı kullanılabilecek steganografi algoritması hakkında önceden bilgi sahibi olmadan steganografinin varlığını tespit etmektedir. "Kör" steganografi tespiti olarak bilinen bu yetenek Stego Suite'e özeldir.

Bir web sitesinde yetkisiz kimselerin zararsız görüntülü mesajlar değiş tokuş etmek veya mülkiyetle ilgili verileri çalmak için siteyi kullanmadıklarından emin olmak için Aktif Web Taraması yapılması giderek bir zorunluluk olmaktadır. Stego Suite araçları bir web etki alanını temel alarak etki alanını sayısal görüntüler içindeki gizli verilerin mevcudiyeti için sürekli biçimde tararlar.

Sayısal bir olay yerinde toplanan verilerin daha sonra analiz edilmesi usandırıcı bir süreç olabilir. Sayısal veri başka gizli kanıt içerebilen binlerce görüntü, ses dosyası veya sayısal video içerdiğinde, kilit teknolojiler ve uzmanlık yardımı olmaksızın süreç imkânsız hale gelebilir. Stego Suite araçları sayısal görüntü ve/veya ses verilerinin gizli bilgiler veya iletişim yönünden incelenmesi ve değerlendirilmesi yeteneğini sağlamaktadır.

Stego Suite 4.1 hem başkasına yaptırılan bir izleme/tarama hizmeti hem de üç uygulamadan oluşan kendi başına bir yazılım paketi olarak kullanımdadır. Seçme yetkisi programın idaresini uzman Wetstone personelinin ellerine bırakmayı arzu eden müşteriler için uygundur. Yazılım paketi steganografi tespit fonksiyonunu kendi sistemleri içinde yapmayı seçen müşteriler içindir. Her iki seçimde de teknik destek ve Stego Suite ekibinin uzmanlığı sağlanmaktadır.

Stego Suite resim ve ses içerisine saklanmış verinin varlığını, saklama algoritmasının bilinmesine ihtiyaç duymadan evrensel olarak tespit edebilen en gelişmiş ticari amaçlı steganaliz yazılımlarından birisidir. BMP, GIF, PNG, JPG ve WAV dosyalarının steganalizini yapabilmektedir [37].

Wetstone Technologies Stego Watch ve Stego Analyst tespit araçları ve bir şifre kırıcı, Stego Break, içeren Stego Suite'i sunmaktadır. Aynı zamanda, Black Hat USA 2005 konferansında diğerleri ile birlikte bu araçları kullanmak için eğitim de önermektedirler.

Bu araçların nasıl steganaliz yaptıkları açık olmadığı gibi hangi steganografi araçlarını tespit ettikleri de bilinmemektedir.

WetStone Stego Suite dört üründen oluşmaktadır:

Stego Hunter™ – Steganografi programı tespiti
Stego Watch™ – WetSyone'nun amiral gemisi olan steganografi tespit aracı
Stego Analyst™ – En son teknolojiye sahip görüntü ve ses dosyası analizcisi
Stego Break™ – Steganografi şifresi kırma

Stego Suite paketi, güvenilir bir laboratuar ortamında yürütülen ve bağımsız yazılımın tespit prosedürlerinin tam kontrolüne izin verdiği, kendini kanıtlamış en yeni steganografi tespit teknolojilerini ve emsalsiz analiz uzmanlığını bir araya getirmektedir. Her durumda seçilen tespit algoritmasının doğrudan veya türetilmiş sonuçlarını ihtiva eden analizi belgeleştiren detaylı bir sonuç raporu verilmektedir.

Stego Suite'in Yararları aşağıdaki gibi göstermektedir:

- Steganografi uygulamalarının hızlı tespiti
- Tespit edilen steganografi programlarının otomatik sınıflandırılması
- Steganografinin anında tespiti
- Görüntü ve Ses dosyalarının detaylı analizi

- Detaylı sonuç raporları
- Yapay olguların görsel tespiti
- Steganografi programlarının kırılması
- Web etki alanlarının sürekli taranması ve izlenmesi
- Yetkisiz bilgi saklamaya karşı aktif savunma

4.4.1. Stego Hunter

Stego Hunter Suite'e en son eklenen özellik olup araştırmacıların şüphe edilen sistemde steganografi programları bulunup bulunmadığını hızla ve dakik olarak tespit etmelerine izin vermektedir. Steganografi uygulamaları bir kez tespit edildiğinde, Stego Hunter sizi tespit edilen programlarla ilişkili taşıyıcı dosyalara otomatik olarak yönlendirecektir.

Araştırma işlemine ilk bakış olarak steganografi programını hızla, dakiklikle ve kolayca tespit etmek için tasarlanmıştır. Çoğu kez bize "steganografinin var olduğunu nasıl biliyorsunuz" diye sorulur. Stego Hunter ile kurulumu yapılan, hatta daha önceden kurulmuş bulunan uygulamaların sonuçları araştırıcıya kolayca rapor edilir. Bu adımı biraz daha ileriye götürüp araştırma sürecinde sonra aramanız gereken bazı şüpheli taşıyıcı tiplerini işaretleriz. EnCase, FTK, dd, raw, ISO ve safeback görüntüler gibi diğer popüler adli araçların adli görüntülerini tarama yeteneğine sahibiz.

4.4.2. Stego Watch

En son teknolojiye sahip aykırılık tabanlı steganografi tespit aracımızdır. Şüphe edilen taşıyıcı dosyalar bir kez bulunduğunda; Stego Watch bütün dosya sistemini otomatik olarak tarayabilir ve sonuçlar şüpheli dosyalar işaretlenmiş olarak okunması kolay bir ara yüze getirilir. İşaretli dosyalar taranmış medya dosyaları içerisindeki yapay olguları arayan bir kör tespit tekniği ile tespit edilir ve daha sonra bulunan yapay olguların tehlike ilişkisi ve bildirimi ile birlikte görüntüye getirilir. Steganografi programları hakkında önceden bilgi sahibi olmak gerekmemektedir.

Stego Watch kullanıcıların sayısal steganografiyi veya sayısal görüntü veya ses dosyalarında saklı iletişimin mevcudiyetini tespit etmelerine izin vermektedir. Stego Watch yazılımının altındaki teknoloji WetStone Technologies'in uzun vadeli ve yoğun araştırma gayretinin sonucu olup, şimdi Stego Suite teklifinin temelini teşkil ederek ticari olarak kullanıma sunulmuştur.

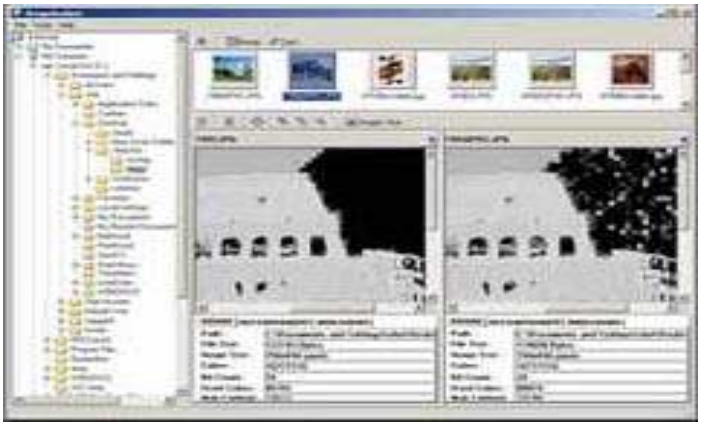
Yeni özellikler geliştirilmiş raporlama ve görüntü tespit algoritmalarını ve. Wav ses formatı tespit yeteneğinin eklenmesini içermektedir. Ek olarak, Stego Watch en popüler steganografi araçlarından bazıları ile yerleştirilmiş bilgileri şimdi bir sözlük saldırısı kullanılarak çıkartabilmektedir.

Araçlara erişmeden onları incelemek mümkün değildir. Stego Watch'u tartışan başka kaynaklar bulunmamıştır, bununla birlikte, yukarıda ifade edildiği gibi, bu araçlar Black Hat USA 2005 konferansında takdim edilecektir.

4.4.3. Stego Analyst

Stego Analyst araştırmacıların ayrıca steganografinin aslında hem görüntü hem ses dosyalarında kullanılmış olduğunu gösteren görsel izleri aramalarına izin veren çok özellikli bir görüntüleme aracıdır. Stego Analyst Stego Watch ile eksiksiz birleşerek şüpheli münferit görüntülerin görsel olarak incelenmesine izin verir. Çoğu kez Stego Watch'ın gömülü steganografi şüphesini teyit etmek için araştırıcıya yardım edenler bu görsel izlerdir.

Şekil 4.5. Normal Görüntünün Görünümü



Şekil 4.6. Steganografi ile veya steganografisiz görüntü tonu [4]

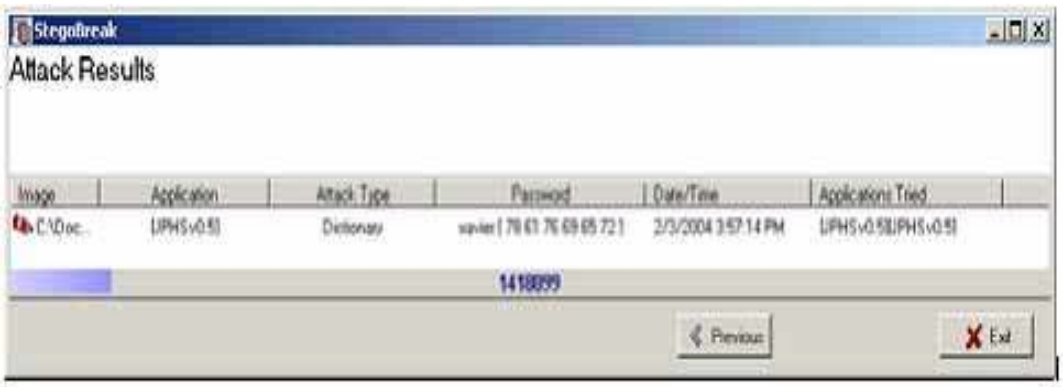
Görsel izlere ek olarak, Stego Analyst kullanıcılara renk paletleri, ton, yoğunluk, kullanılan renkler ve kırmızı-yeşil-mavi (RGB) değerleri gibi özellikleri inceleme yeteneğini vermektedir.

Araştırıcıların steganografinin görüntü ve ses dosyalarında fiilen kullanılmasına ait görsel izleri aramalarına izin veren çok özellikli bir görüntüleme ve analiz aracıdır. Münferit dosya görüntüsü veya ses dalgasını görüntüleyen bir dosya seyretme ekranı ve görüntü detayları, DCT katsayıları, renk çiftleri vb. içeren dosya nitelikleri sağlanmıştır. Araştırıcıların steganografinin kullanılmakta olduğu hakkında daha fazla iz aramalarına izin vermek için, görüntüleri Yoğunluk, Doygunluk veya Ton şeklindeki üç değişik sunumdan birine dönüştüren filtre seçme yetkililerini dahil ettik. Diğer filtre seçme yetkilileri sadece seçilmiş özel renkli Least Significant Bit (LSB)leri görüntülemektedir. Birçok steganografik teknik veri gizleme için LSB'leri kullandığından, bir görüntünün LSB'lerini seyretmek bazen steganografi göstergelerini açığa çıkarabilmektedir.

4.4.4. Stego Break

Steganografi içerdiği bulunan bir dosyada kullanılmış olan parola ibaresini elde etmek için tasarlanarak yerleştirilmiş bir programdır. Aracın satın alınması ile bir sözlük saldırısı icra etmek için popüler parola sözlükleri de verilmektedir.

Araştırıcılar aynı zamanda başka sözlükler getirme yeteneğine sahip olup, eğer şüphe sorgulaması vasıtasıyla parolayı bulurlarsa tespit edilen görüntü veya ses dosyalarına karşı parolayı kullanabilirler. Şifrelenmiş bir dosyayı bulup çıkartabilirseniz, bir şifreleme uzmanı ile temas kurmanızı teşvik edebiliriz.



Şekil 4.7. Stego Break

4.5. Steganalyzer

Çizelge 4.7. StegAnalyzer

Adı	StegAnalyzer
Müellifi	Backbone Security
Lisans durumu	Lisanslı
Tespit edilen yazılım	Bilinmiyor
URL	www.sarc-wv.com/products.aspx

Backbone Security StegAnalyzer'ın üç sürümünü sağlamaktadır. StegAnalyzer AS bilinen steganografi yazılımının izleri için dosya sistemlerini araştırır. StegAnalyzer SS bilinen stego dosya imzalarını tespit etme işlevselliğini içermektedir. StegAnalyzer RTS parmak izlerini ve imzalarını gerçek zamanlı olarak kullanılabilir ağ uygulamasıdır.

Bu yazılımın bir kopyası mevcut değildir, ancak hala bazı düşünceler öne sürülebilir. StegAnalyzer bir hayli pahalıdır; yaklaşık $2000 bedelle satılmaktadır. Bununla

birlikte, işlevselliği diğer araçlardan elde edilebilir. Dosya bozuk imzalarının tespiti, aynı şekilde bilinen stego dosyası imzaları, ücretsiz araçlarla desteklenmektedir (örneğin Sleuthkit/Autopsy). Böylece StegAnalyzer'ın gerçek değeri yazılım ve stego dosyası imzaları ile veri tabanındadır. Bu veri tabanının kalitesi bilinmemektedir.

4.5.1. Steganografi Analizi Yapay Olgu Tarayıcısı (StegAlyzerAS)

StegAlyzerAS (Steganography Analyzer Artifact Scanner) steganografi uygulamalarındaki bilinen yapay olgular için inceleyicinin şüphe edilen ortam veya şüphe edilen ortam görüntülerini taramasına izin vererek geleneksel sayısal adli incelemelerin kapsamını genişletmek için tasarlanmış bir sayısal adli analiz aracıdır.

Artık yapay olgular dosya sistemi, aynı şekilde Microsoft Windows® sistemi kayıtları taranarak tespit edilebilmektedir. StegAlyzerAS birlikte verilen Steganografi Uygulaması Parmak İzi Veri Tabanında (SAFDB) saklanan CRC-32, MD5, SHA-1, SHA-224, SHA-256, SHA-384, ve SHA-512 karma değerleri ve Yapay Olgu Kayıt Ana Veri Tabanında (RAKDB) saklanan kayıt girişleri kullanılarak dosyaların aranmasına izin vermektedir.

StegAlyzerAS'da öne çıkan ürün özellikleri:

- Olay üretim ve yönetimi (Yüklemsiz Tümce).
- EnCase, RAW (dd), veya SMART formatlarında depolama medyasının adli görüntülerini monte etme ve tarama yeteneği.
- Şüpheli medyadaki bütün dosya sistemi, münferit dizinler veya münferit dosyaların steganografi uygulaması dosya yapay olgularının mevcudiyeti için otomatik olarak taranması.
- Microsoft Windows® Registry sisteminin belirli steganografi uygulamaları ile ilişkili kayıt yapay olguların mevcudiyeti için otomatik olarak taranması.

- Dosya ve kayıt yapay olgu delil görüntüleyicileri tespit edilen her steganografi uygulamasında keşfedilen yapay olgu yüzdesine göre inceleyicinin delilleri görüntülemesine izin vermektedir.
- Tarama özet görüntüleyici inceleyicinin belirli bir inceleme sırasında yapılan önceki bir taramanın istatistiksel özetini hızla görmesine izin vermektedir.
- HTML formatında kapsamlı rapor üretimi.
- Potansiyel delil değeri olan bilgilere ait esas olaylar ve bilgilerin otomatik kaydı.
- Özgün özellikler ve işlevleri açıklamak için bütünleşmiş yardım özelliği.

StegAlyzerAS'nin gelecekteki bir sürümüne eklenecek özellikler:

- Ek steganografi uygulamaları bulundukça ve bu uygulamalarla ilişkili yapay olgular SAFDB ve RAKDB'ye eklendikçe, genişletilmiş yapay olgu tarama yeteneği
- Uygulamaya ait yapay olgular tespit edildiğinde inceleyiciye uygulama hakkında ek bilgi vermek üzere SAFDB içindeki steganografi uygulamalarının görüntü yararları da aşağıda göstermektedir:
- Sayısal steganografi uygulamalarındaki yapay olguların araştırılması
- Şirketinizin fikri mülkiyetinin korunması
- Güven duyulmuş olan içten öğrenicilerin şirket ağınızdan dışarıya hassas ve/veya özel bilgiler göndermek için gizli teknikler kullandığının belirlenmesi
- Sayısal steganografi veya diğer veri gizleyici uygulamaların kullanımını engelleyen kurumsal politikaların yürürlüğe sokulması
- StegAlyzer AS'e kayıt taraması özelliğinin inceleyicilerin Microsoft Windows® Registry içindeki steganografi uygulamalarına ait yapay olguları tespit etmelerine izin vermesi
- Hemen her yerde aleni olarak erişimi sağlanabilen en yaygın steganografi uygulama kurguları kullanılarak dosya yapay olgularının araştırılması
- Yedi değişik kurma algoritmasından her hangi biri kullanılarak dosya yapay olgularının doğrulanması

4.5.2. Steganografi Analizi İmza Tarayıcısı (StegAlyzerSS)

StegAlyzerSS (Steganography Analyzer Signatures Scanner) içlerine gizli bilgiler gömmek için belirli steganografi uygulamaları kullanıldığında, StegAlyzerSS inceleyenin şüpheli medyadaki dosyaları veya şüpheli medyanın adli görüntülerini, dosyaların içinde kalmış ve benzersiz onaltılık bayt modelleri veya bilinen imzalar için taramasına izin vererek geleneksel adli incelemelerin kapsamını genişletmek üzere tasarlanmış bir sayısal adli analiz aracıdır.

StegAlyzerSS imza tarama yeteneğini ayrıca inceleyicinin bilgilerin potansiyel taşıyıcı dosyalara eklenmiş veya içerisinde gizlenmiş olabileceğini tespit etmek için başka teknikler kullanmalarına izin vererek de genişletebilir.

Defense Cyber Crime Institute (DCCI) ve CyberScience Laboratory (CSL) StegAlyzerSS'nin gizlenmiş steganografik verileri içeren dosyaları tespit etmekte etkili olduğunu ortaya koymuştur [93,94]. StegAlyzerSS'de öne çıkan ürün özellikleri:

- Olay üretim ve yönetimi.
- EnCase, RAW (dd), veya SMART formatlarında depolama medyasının adli görüntülerini monte etme ve tarama yeteneği.
- Şüpheli medyadaki bütün dosya sistemi, münferit dizinler veya münferit dosyaların belirli steganografi uygulamalarının bilinen imzalarının mevcudiyeti için otomatik olarak taranması.
- Appendx Analysis özelliği ile dosya bitim işaretleyicisinin ötesine eklenmiş bilgilerin bulunduğu dosyaların tespit edilmesi ve gizlenmiş bilgilerin niteliğini belirlemek için dosyaların onaltılık bir editör görüntülemesinde analiz edilmesi.
- LSB Analysis özelliği ile Least Significant Bit (LSB) görüntü kodlaması kullanılarak gömülü bilgilerin bulunduğu dosyaların tespit edilmesi ve LSB'lerin dosyanın içerisinde bilgi gizlenip gizlenmediğinin belirlenmesi için onaltılık bir editör görüntülemesinde analiz için çıkartılmaları ve yeniden düzenlenmeleri.

- Seçilmiş steganografi uygulamaları için Özel Otomatik Çıkartma Algoritması işlevselliği inceleyicilere şüphe edilen dosyalardan gizli bilgileri kolayca çıkartmak için bir "işaretle-tıkla-çıkart" ara yüzü sağlamaktadır.
- HTML formatında kapsamlı rapor üretimi.
- Potansiyel delil değeri olan bilgilere ait esas olaylar ve bilgilerin otomatik kaydı.
- Virgülle ayrılan değer (.csv) formatında dışarı aktarılan oturum faaliyeti ve delil kayıtları.
- Özgün özellikler ve işlevleri açıklamak için bütünleşmiş yardım özelliği.

StegAlyzerSS'nin gelecekteki bir sürümüne eklenecek özellikler:

- Genişletilmiş imza tarama ve tanıma yeteneği.
- Genişletilmiş kör tespit analizi yeteneği.
- Ek Otomatik Çıkartma Algoritmaları.

Yararları da aşağıdaki gibi sıralanabilir:

- Gizlenmiş bilgilerin başka türlü fark edilmeyen cezai faaliyetlerin delili olarak kullanılmak üzere keşfedilmesi için dosyaların taranması.
- Gizli iletişimin delillerinin keşfedilmesi.
- Kurumunuzun fikri mülkiyetinin ve/veya diğer hassas veya özel bilgilerin korunması.
- Güvenilen içeriden öğrenenlerin şirket ağınızdan dışarıya hassas ve/veya özel bilgileri göndermek için gizli teknikler kullanıp kullanmadığının belirlenmesi.
- Sayısal steganografi veya diğer veri gizleme uygulamalarının kullanılmasını yasaklayan kurum politikalarını yürürlüğe sokmak.
- Otomatik Çıkartma Algoritmaları inceleyicilerin StegAlyzerSS'ye özgü bir özellik olan "işaretle-tıkla-çıkart" ile gizlenmiş bilgileri bulmalarına izin verir.

4.5.3. Steganografi Analizi Gerçek Zamanlı Tarayıcısı (StegAlyzerRTS)

StegAlyzerRTS (Steganography Analyzer Real-Time Scanner) sayısal steganografi uygulamalarının parmak izlerini ve imzalarını gerçek zamanlı olarak dünyada tespit etme yeteneğine sahip ilk ticari olarak kullanılabilir ağ güvenliği uygulamasıdır.

Steganografi uygulamalarına Internet üzerinde serbest veya paylaşımlı biçimde yaygın olarak erişilebilir ve bunlardan bazıları ticari lisanslı ürünler olarak kullanılabilir. Bunlar önemli bir içeriden öğrenme tehdidini ifade ederler çünkü birçok kullanıcı uygulamasındaki bilinen ve mevcut "sürükle ve bırak" işlevselliği ile bunları bulmak, indirmek, kurmak ve kullanmak kolaydır.

StegAlyzerRTS içeriden öğrenici indirilmesini veya sayısal steganografi uygulamaları kullanılmasını dosya parmak izlerini veya karma değerlerini, steganografi uygulamaları ile ilişkili dosyaların parmak izleri ile karşılaştırarak tespit etmektedir. Bir indirmeyi tespit etme içeriden öğrenicinin bilgiyi başka bir dosyanın içine gizleyerek ve sonra taşıyıcı dosyayı bir web sitesine postalayarak veya taşıyıcı dosyayı e-posta ile harici bir alıcıya göndererek hassas, özel, fikri mülkiyeti çalmak veya gizli bilgilerin ticaretini yapmak için muhtemelen uygulamayı kullanacağını önceden gösteren bir uyarıdır.

StegAlyzerRTS ayrıca StegAlyzerRTS konuşlandırılmadan önce indirilmiş ve ağa kurulmuş steganografi uygulamalarının içeriden öğrenici kullanımını da tespit etmektedir. Steganografi Analiz ve Araştırma Merkezi'nde (SARC) geliştirilmiş özel bir imza tarama yaklaşımı kullanılarak, StegAlyzerRTS içeriden kullanıcının gizli bilgileri içeren taşıyıcı dosyaları harici web sitelerine yükleme, gizli bilgileri içeren taşıyıcı dosyaları e-posta eklentisi olarak gönderme teşebbüslerini, hatta bilgiyi spam görünümünde bir şekle dönüştürerek gizlemek için spam taklitçiliği olarak bilinen bir tekniğin kullanılmasını bile tespit etmektedir.

Aşağıda StegAlyzerRTS'de öne çıkan ürün özelliklerinin bir listesi bulunmaktadır:

- Ağa giren ve çıkan sayısal steganografi uygulamaları ile ilişkili yapay olguların tespit edilmesi.
- Ağa giren ve çıkan taşıyıcı dosyalardaki steganografi uygulamalarının imzalarının tespit edilmesi.
- Ağ güvenlik yöneticilerine gerçek zamanlı uyarılar gönderilmesi.
- Şüphe edilen dosyaların şirket ağından giriş veya çıkışına izin vermek veya geri çevirmek.
- Daha çok analiz için şüphe edilen dosyaların kopyalarının tutulması.

StegAlyzerRTS kullanışlı ve dertsiz bir "parazit karakter" kapasitesi sunmaktadır. İşletmenin mevcut güvenlik uygulamaları ve araçları takımına kolayca eklenebilir ve son kullanıcıya karşı tamamen şeffaftır.

StegAlyzerRTS'yi hemen konuşlandırarak mevcut hassas bilgi ve fikri mülkiyet Koruma mekanizmalarını arttırın!

Yararları da aşağıda göstermektedir:

- Dosya parmak izlerinin – dosyaların steganografi uygulamaları ile ilişkili karma değerlerinin veya yapay olguların – gerçek zamanlı tespiti.
- İmzaların – taşıyıcı dosyalardaki onaltı baytlık örüntülerin – gerçek zamanlı tespiti.
- Güvenlik yöneticilerine gerçek zamanlı uyarı
- Gelen veya giden şüpheli dosyaların karantinaya alınması veya geçiş izni verilmesi.
- Sayısal steganografi veya diğer veri gizleyici uygulamaların kullanımını engelleyen kurumsal politikaların yürürlüğe sokulması.
- Hassas bilgilerin ve fikri mülkiyetin sızmasına karşı koruma.
- İçeriden öğrenicilerin indirilmesinin ve hassas ve/veya özel bilgileri şirket ağından dışarıya göndermek için steganografi uygulamalarının kullanılmasını tespit etme.

4.6. Karşılaştırma Ve Sonuç

Bu bölüm steganaliz için dört ve steganografiye karşı sözlük saldırısı için bir aracı ele almıştır. Bunların ikisi lisanslıdır; bu yüzden bu yazılımlar elde edilmemiştir. İki parasız steganaliz aracı, StegSpy ve Stegdetect, değişen sonuçlarla test edilmiştir. StegSpy dosya imzalarını hedef almıştır, yani bilinen onaltılık sistemin (hexadecimal) değerlerini aramaktadır, öte yandan Stegdetect'in daha karmaşık bir yaklaşımı vardır. StegSpy'ın bir sınırlaması tespit edilmiş ve dosya aykırılıklarının araştırılmasını içeren iyileştirilmiş bir çözüm önerilmiştir.

Bu bölümde çizelge 4.8'de mevcut yazılımlar karşılaştırılmalı olarak verilmiştir.

Çizelge 4.8. Steganaliz Uygulamalar

Ürün Adı	StegSpy	Stego Suite	Stegbreak	StegAnlyzer	stegdetect
Kodu	Açık kaynak	Lisanslı	Açık kaynak	Lisanslı	Açık kaynak
Hangi program diliyle yazılmış	Visual Basic	Bilinmiyor	Visual Studio.NET	Bilinmiyor	Visual Studio.NET
Geliştiren	InfoSec 2004, BlackHat 2004 and DefCon 2004	Wetstone Technologies	Niels Provos	Backbone Security	Niels Provos
Kullanılan Formatları	JPHideandSeek, Hiderman , JPegX,Invisible Secrets gibi birkaç steganograilk yazılımla oluşturan Stego dosyaları tespit etmektedir (Sadece resimler)	Resim, Bmp,gif,png,jpg Ses, wav	JSteg-Shell, JPHide and OutGuess 0.13b (Sadece jpg resimler)	Bilinmiyor	jsteg, jphide, invisible secrets, Outguess 0.13b, F5, appendX and camouflage. (Sadece jpg resimler)

Çizelge 4.8 (Devam) Steganaliz Uygulamalar

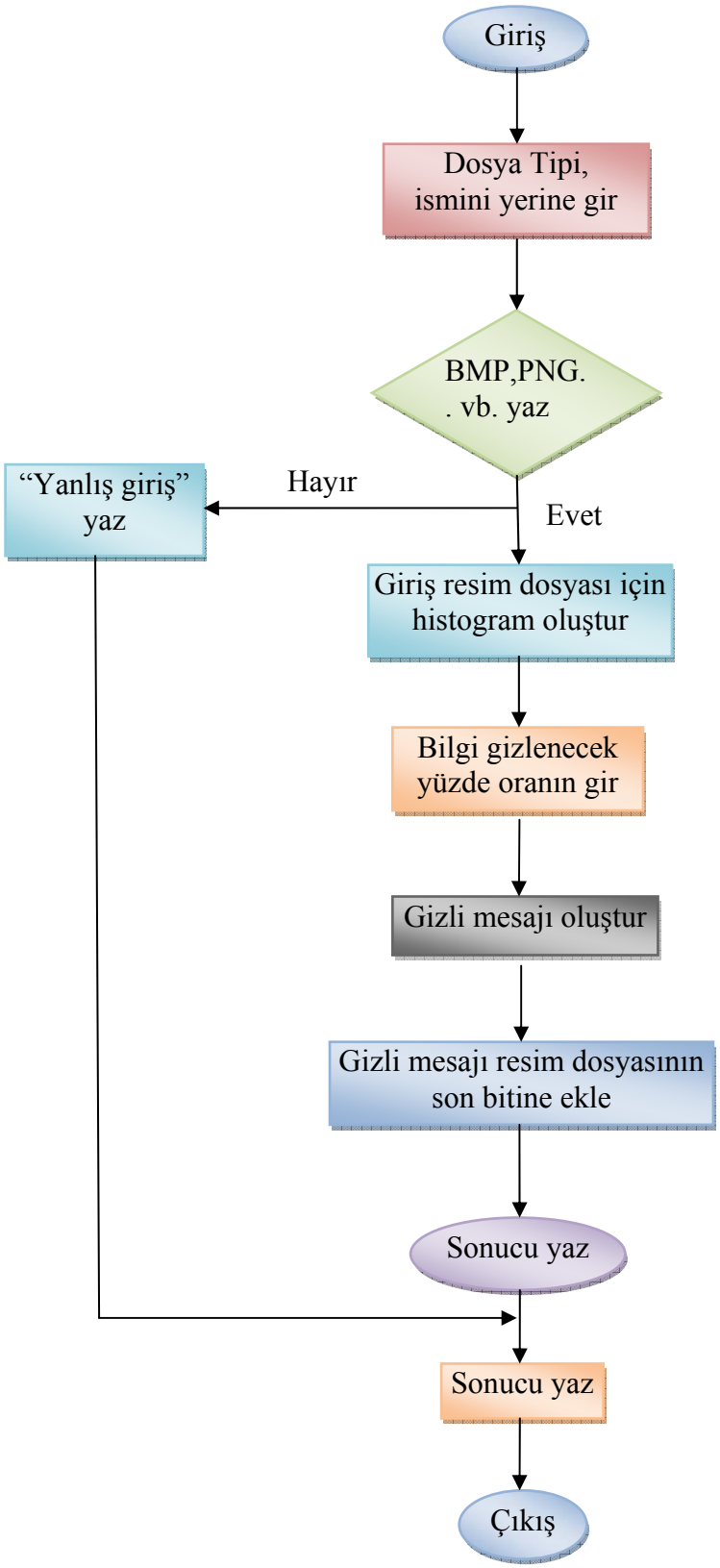
Deneyim Türü	Dosyada veri saklı olup olmadığını ve saklıysa hangi yazılımla saklandığını tespit etmektedir	Evrensel tespit	Algoritmaya özel tespitler	Evrensel tespit	Algoritmaya özel tespitler
www Adresler	www.spyhunter.com/ stegspydownload.htm	www.wetstonetech. com/	www.outguess.org/ detection.php	www.sarc-wv.com /products.aspx	www.outguess.org/ detection.php
Desteklenen hizmetler	Bilinmiyor	Wetstone Technologies	Wetstone Technologies	Backbone Security	Bilinmiyor
Güncelleme durumu	YOK	Stego Hunter Stego Watch Stego Analyst Stego Break	YOK	StegAnlyzerAS StegAnlyzerSS StegAnlyzerRTS	OutGuess 0.2 - 2001-02-12 Stegdetect 0.2 x- 2001–07–23 Stegdetect 0.3 - 2001-10-02 Stegdetect 0.4 - 2001-12-21 Stegdetect 0.5 - 2002-01-26 Stegdetect 0.6 - 2004-09-06

5. RESİM TABANLI STEGANALİZ UYGULAMALARI

Steganografinin alt disiplinlerinden biri, belirli bir örtü ortamının gömülü bilgi içerip içermediğinin belirlenmesini kapsayan steganalizdir. Örtü görüntünün gizlenmiş veri içerip içermediğini belirlemeye ek olarak, steganaliz ayrıca mesajın uzunluğu (yani gömülen bitlerin sayısı), örtü görüntüde mesajın nerede olduğunu, mesajı gizlemek üzere hangi tekniğin kullanıldığı ya da mesajın fiili içeriği gibi gizlenmiş veriler hakkında bir şeyler de öğrenmeye çalışır. Literatürde önerilen steganaliz tekniklerinin çoğu LSB gömme teknikleri kullanılarak gömülen veri içeren ve içermeyen görüntüler arasında ayrım yapmak üzere tasarımlanmasına rağmen, LSB teknikleri ile gömülen mesajlar hakkında daha fazla bilgi öğrenmek ya da LSB olmayan gömme algoritmalarına saldırmak üzere bir kaç yeni teknik geliştirilmiştir.

Bu bölümün hedefi Westfeld ve Pfitzmann'ın Ki-kare saldırısıyla [58] deneyler yapmak ve ideal olarak sonuçlarını doğrulamaktır. Asıl amaç, bu araştırmada söz edilen algoritmalardan en azından birini kullanılarak gömülmüş olan veriler içeren görüntüler üzerinde testlerini tekrarlamaktır. Bununla birlikte, belirtilen algoritmaların çoğu artık orijinal sürümlerinde mevcut değildir. Dahası, hala mevcut olan algoritmalar Ki-kare saldırısı kullanılarak algoritmayı analiz etmenin gerekli bir bileşeni olan gömme sürecinin nasıl çalıştığını tanımlayan belgelerden yoksundur. Ne var ki, Westfeld ve Pfitzmann Ki-kare saldırısının diğer gömme algoritmalarına başarılı bir şekilde saldırmak üzere değişikliğe uğratılabilir; dolayısıyla üzerinde Ki-kare saldırısını test edeceğim bir algoritma oluşturdum.

Bu çalışmada kullanılan algoritma Şekil 5.1'de verilmiştir.



Şekil 5.1. Resim tabanlı bilgi gizleme akış diyagramı

5.1. Kararlar Ve Varsayımlar

Bu algoritmaları oluşturmada, bir kaç karar aşağıdaki şekilde alınmıştır:

1. Gömme algoritması piksel başına en fazla bir bit gömecektir. Bazı pikseller gömülü bitler içermeyebilir. Bu karar hem basitlik açısından hem de veriyi görüntü içerisine algılanamaz bir şekilde gömmek için alınmıştır.

2. Gömme algoritması bitleri, örtü görüntünün üst sol köşesinden başlayarak ilk sıradaki, daha sonra ikinci sıradaki piksellere ve örtü görüntüye her bir mesaj biti gömülene Kadar ardışık bir şekilde gömecektir. Bu gömme planının amacı büyük oranda süreci hata ayıklama için basitleştirmektir ancak aynı zamanda bir görüntünün çeşitli kısımlarını yani veri gömülü kısımlar ve veri gömülmemiş kısımları test etmemize izin vermektedir.

3. Her bir mesaj gömme işleminden önce rastgele olarak oluşturulacaktır. Bu tasarımın birinci amacı sonuçlarımı gömülmüş mesajın özel içeriğine atfetme olasılığını en aza indirgemektir. Eğer gömme işlemi her meydana geldiğinde mesaj değişirse ve her yeni mesaj gömüldüğünde sonuçlar uyumluysa sonuçlar doğrudur.

Algoritmaları oluşturmada, bir kaç varsayım aşağıdaki şekilde alınmıştır:

1. Mesajdaki bitlerin sayısı örtü görüntüdeki piksellerin sayısına eşit ya da daha düşüktür. Bu durum, her bir pikselin en fazla bir adet gömülü biti olduğu kararını karşılamak üzere gerekli bir koşuldur.

2. Mesaj 0 ve 1'lerin rastgele dağılımıdır. Genel olarak steganografi amacıyla görüntülere gömülen veriler gömülmeden önce şifrelenerek, böylece rastgele bir 0'lar ve 1'ler dağılımı oluşturur. Dolayısıyla gömülmekte olan verilerin şifrelenmiş olduğunu varsayıyoruz.

5.2. Veri Saklama Yaklaşımı (FE aracı)

Yüksek düzey bir açıklama olarak, yazdığım FE aracı adlı algoritma mesaj bitine uyacak şekilde örtü görüntüdeki her bir pikselin LSB'sini "tersine çevirmektedir". $c_{ij} \in \{0, 1, 2, ... , 255\}$ elemanlarıyla $C^{p\times q}$ 8 bitlik gri ölçekli örtü görüntüsünü temsil etsin; $m_{ij} \in \{0, 1\}$ elemanlarıyla $M^{p\times q}$ gizli mesajı temsil etsin ve $s_{ij} \in \{0, 1, 2, ..., 255\}$ elemanlarıyla $S^{p\times q}$ (8 bitlik gri ölçekli bir görüntü olan) stego görüntüyü temsil etsin. C örtü görüntüsü kullanıcı tarafından seçilmiş ve FE araca okutulmuştur. Varsayılan olarak, Matlab görüntüleri matrisler olarak okur ve 8 bitlik matrisler için her bir elemanın 0 ila 255 arasındaki tam sayılardır. Görüntüyü okuduktan sonra, M rastgele dağıtılmış 0'lar ve 1'lerden oluşmuş bir matris olacak şekilde ilk olarak 0 ila 1 arasında değişen rastgele oluşturulmuş p × q'luk matris olarak M' matrisini oluşturur ve daha sonra M= mod([10M'], 2)'yi hesaplayarak FE aracı mesajı oluşturur. Kullanıcı ayrıca örtü görüntünün büyüklüğünün % 1'i ila % 100 arasında değişebilecek bir mesaj uzunluğu da seçebilmektedir. Bu özelliğin amacı hem değişmiş hem de değişmemiş piksel değerleri içeren görüntüler üzerinde Ki-kare saldırısını test etmemizi sağlamaktır. Her bir piksele veri gömmenin steganografi sistemi kapasitesi için makul beklentileri aşması iddia edilebileceğinden dolayı piksellerin % 100'ünden daha azına bilgi gömülü olan görüntüleri test etmek faydalı olacaktır. Dahası bir imaja gömülü durumda olan mesaj pratik durumlarda nadiren örtü görüntünün % 100'ünü tam olarak dolduracaktır. C ve M'i elde ettiğimizde stego görüntü S aşağıdaki şekilde oluşturulacaktır:

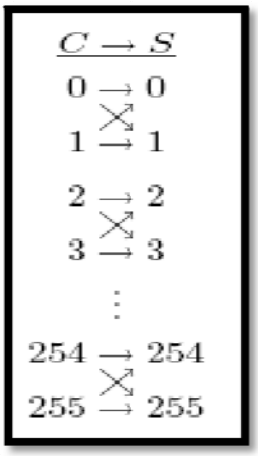
l', $l' \in \{0.01, 0.02, ..., 0.99, 1\}$ olacak şekilde mesaj bitlerini gömeceğimiz piksellerin ondalık sayı cinsinden temsili olsun. Daha sonra, l gömülecek bitlerin sayısı olacak şekilde $l = [l'pq]$ yazalım. $l = (m - 1)q + n$ ve $1 \le m \le p$, $1 \le n \le q$ olsun. c_{mn} verinin gömüleceği örtü görüntünün son elemanıdır. m ve n üzerindeki kısıtlamalar ve önceden belirlenmiş veri gömme sıralaması nedeniyle c_{mn} l ve q tarafından benzersiz bir şekilde belirlenmektedir. Artık S' = 2(C/2) + M olsun. Bu işlem piksel değerlerinin en yüksek 7 bitini değiştirmeden bırakmakta ve mesaj bitine uyacak şekilde (gerekiyorsa) LSB (en az önemli biti) değiştirir. Eğer l = pq ise, S = S'dır.

Bununla birlikte eğer $l < pq$ ise, aşağıdakinin stego görüntü olması için $s_{11} = s'_{11}$, ..., $s_{1q} = s'_{1q}$, ... , $s_{mn} = s'_{mn}$, $s_{m(n+1)} = c_{m(n+1)}$, ... , $s_{mq} = c_{mq}$, ..., $s_{pq} = c_{pq11}$ olsun.

$$S = \begin{bmatrix} s'_{11} & \cdots & \cdots & \cdots & \cdots & s'_{1q} \\ \vdots & & & & & \vdots \\ s'_{(m-1)1} & \cdots & \cdots & \cdots & \cdots & s'_{(m-1)q} \\ s'_{m1} & \cdots & s'_{mn} & c_{m(n+1)} & \cdots & c_{mq} \\ c_{(m+1)1} & \cdots & \cdots & \cdots & \cdots & c_{(m+1)q} \\ \vdots & & & & & \vdots \\ c_{p1} & \cdots & \cdots & \cdots & \cdots & c_{pq} \end{bmatrix}$$

Böylece stego görüntü S' 'ün C'nin son pq-*l* piksel değerleri eklenmiş (gömme sıralamasını takip eden) ilk *l* piksel değerleridir. FE aracı yalnızca en az önemli biti değiştirmesi ve her bir piksel değerinin en yüksek 7 bitini S ve C'dekiyle aynı bırakması nedeniyle, {0, 1} → {0, 1}, {2, 3} → {2, 3}, {4, 5} → {4, 5}, ... , {254, 255} → {254, 255} olacak şekilde (Şekil 5.1) bir eşleme sahip oluruz ve stego görüntünün en az önemli bitleri mesaj bitine karşılık gelir. Dolayısıyla mesaj stego görüntünün en az önemli bitini okuyarak kolaylıkla alınabilir, yani M = mdo(S, 2).

FE aracı gibi bir gömme algoritmasının sonuçlarından biri birbirine gömülen piksel değerlerinin, değer çiftlerinin (DÇ) oluşturulmasıdır. Örneğin örtü görüntüsündeki 100 piksel değeri ya 100 olarak kalacak ya da 101'e değişecektir. Benzer bir şekilde 101'lik piksel değeri ya 101 olarak kalacak ya da 100'e değişecektir.



Şekil 5.2. FE aracıdaki olası eşleşmeler.

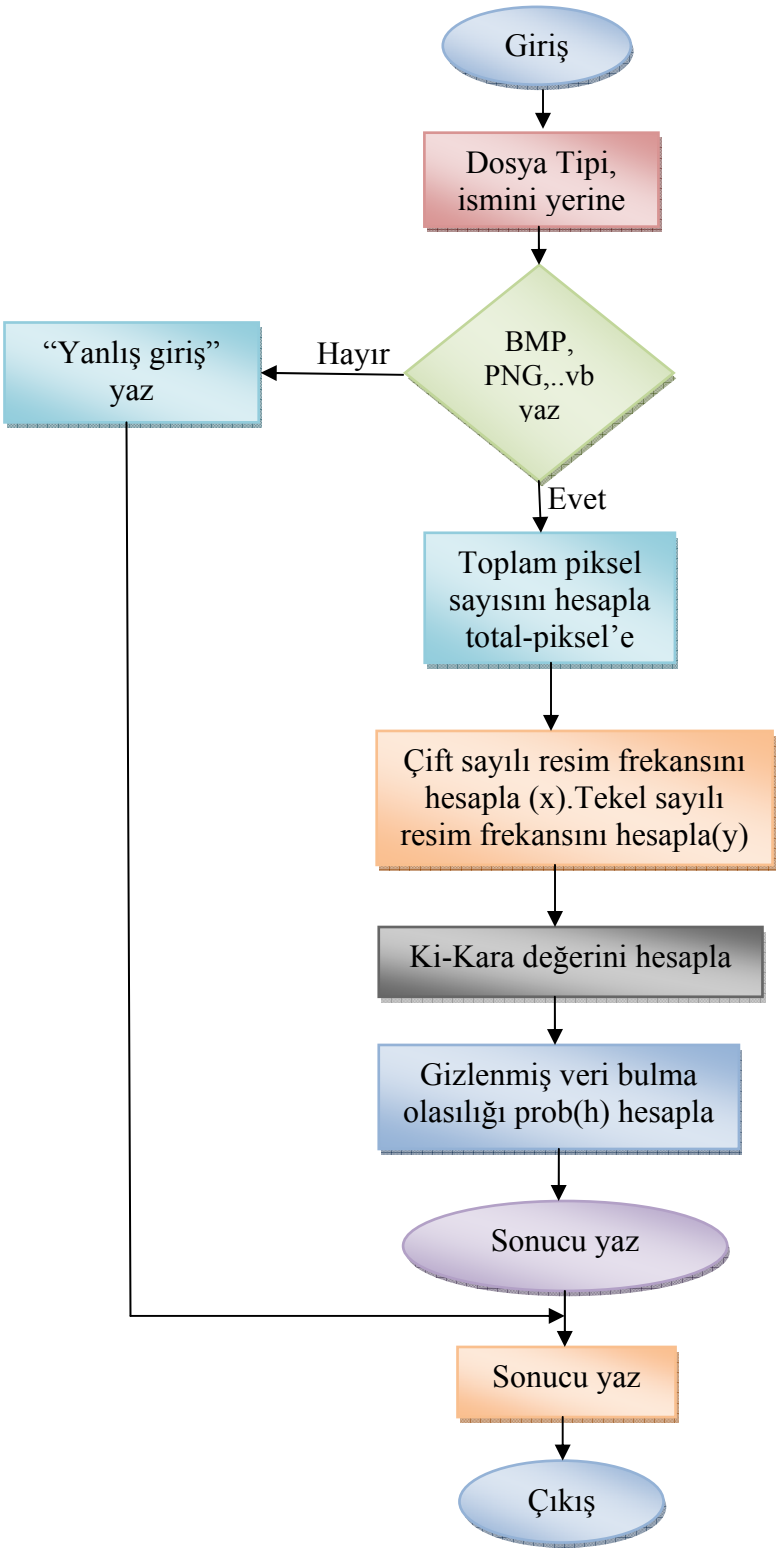
Böylece {100, 101} bir değer çiftidir. Genel olarak, {2k, 2k + 1 | 0 ≤ k ≤ 127} değer çiftlerini oluşturur.

5.3. Ki-kare Saldırısı

Makalelerinde, Westfeld ve Pfitzmann [58] geleneksel yapısının aksine, bir görüntüdeki en az önemli bitlerin tam olarak rastgele olmadığını iddia etmektedir. Dahası, her bir değer çiftindeki iki pikselin her birinin sıklığının değer çiftinin ortalamasından uzağa düşme eğilimde olduğunu iddia etmektedirler. Diğer bir deyişle, içine veri gömülmüş tipik bir görüntü içerisinde 2k piksel değerinin sıklığının 2k + 1 piksel değerinin sıklığına (neredeyse) yakın olması yaygın olmayan bir durumdur. Dahası, bilginin değer çiftleri yaratan FE aracı gibi bir algoritma kullanılarak bir görüntüye gömülmesi nedeniyle 2k ve 2k + 1 eşit ya da neredeyse eşit hale gelir. Ki-kare saldırısı bu neredeyse eşit değer çiftlerini algılamak üzere tasarımlanmıştır ve gömme olasılığını test görüntüsündeki çift sayılı piksel değerleri ve bunların karşılık gelen tek sayılı piksel değerlerinin eşite ne kadar yakın olduğuna dayandırmaktadır.

Ki-kare saldırısı çeşitli gömme algoritmalarına uyarlanabilir olarak tasarımlanmıştır ancak temel kavram gömme algoritmasına bakılmaksızın aynıdır. Bölüm 5.3.1. FE aracı uygulandığı şekilde Ki-kare saldırısını tanımlamaktadır.

5.3.1. Saldırıda kullanılan DÇ3 Algoritması



Şekil 5.3. Resim tabanlı bilgi çıkarma akış diyagramı

DÇ3 algılama algoritması, bir görüntü üzerine Ki-kare saldırısı gerçekleştirip veri gömme olasılığını çıktı olarak vermek üzere tasarımlanmıştır. Kullanıcı bir görüntü seçmekte ve komut isteminde dosya adını ve konumunu girmektedir. DÇ3 piksellerin her bir tam yüzdesini % 1'den % 100'e kadar test etmekte ve test görüntüsü içerisindeki her bir piksel alt kümesi için hesaplanan olasılıkları (Eşitlik 2) geri döndürmektedir. Pikseller, bitler gömülürken FE aracı'nın görüntü içerisinde takip ettiği yolla aynı yolu takip ederek sol üst köşeden yani (1,1) indeksinden başlayarak ardışık olarak test edilmektedir. DÇ3'ün FE aracı'nın bilgiyi gömme sıralamasıyla aynı sıralamada piksel gruplarını test etmesi rastlantısal değildir. İlk olarak daima steganografik algoritmanın saldırgan açısından bilinmediği genel olarak varsayılmalıdır. İkinci olarak ise Westfeld ve Pfitzmann, saldırıda bulundukları steganografik sistemler hakkında bilgiler varsaymışlar ve algılama algoritmalarını buna uygun olarak değiştirmişlerdir; dolayısıyla onların araştırmasını mümkün olduğu kadar en yakın tekrarlamak üzere ben de aynısını yaptım.

DÇ3'ün uygulaması aşağıdaki gibidir:

x_k = (2k) sıklığı ve y_k (2k + 1) sıklığı, $0 \leq k \leq 127$ olacak şekilde X^{128x1} ve Y^{128x1} iki vektör olsun. İlk olarak X ve Y'deki her bir eleman 0'a ayarlanır. Daha sonra DÇ3 test görüntüsü içerisindeki gri değerleri sayar ve X ya da Y'deki karşılık gelen elemanı artırır. Teorik olarak 2k ve 2k + 1'in gri değerlerinin beklenen sıklığı z_k = (xk + yk) / 2'dir. Şimdi n adet kategori olduğunu yani n değer çifti olduğunu farz ediniz. 8 bitlik gri görüntüler olması durumunda, 128 kategori (256 gri değer / 2) vardır. Genelliğini kaybetmeksizin, k kategorisinde ölçülen meydana gelme sıklığı xk olacak şekilde değer çiftlerinin çift değerleri üzerinde odaklanacağız. (Bir örtü görüntüsü ve stego görüntü içerisindeki 2k ve 2k + 1 piksel değerlerinin sıklığının toplamlarının aynı olduğuna yani bir örtü görüntü içerisindeki xk + yk'nın ve karşılık gelen stego görüntü içerisindeki xk + yk'ya eşit olduğuna dikkat ediniz. Daha sonra Westfeld ve Pfitzmann minimum sıklık koşulu ortaya koymuştur, dolayısıyla eğer $0 \leq k \leq 127$ Aralığı için $x_k + y_k \leq 4$ ise, $x_k = y_k = z_k = 0$ ve n = n - 1'dir. Diğer bir deyişle 2k ve 2k+1'in birleşik sıklığı 4'ten az ya da eşitse, 2k ve 2k + 1'in bireysel

sıklık sayımları 0'a ayarlanır ve n kategorisi sayısı 1 azaltılır. Daha sonra n - 1 serbestlik derecesiyle Ki-kare istatistiği hesaplanır.

$$X_{n-1}^2 = \sum_{i=0}^{127} \frac{(x_i - z_i)^2}{z_i}, \qquad \text{where } z_i = \frac{x_i - y_i}{2} \tag{5.1}$$

Bir stego görüntü için x_i'nin z_i'ye yakın olması gerektiğinden dolayı X^2_{n-1}'in görece küçük olması ve x_i'nin z_i'den uzak olması gerektiğinden dolayı X^2_{n-1}'in görece büyük olması beklenir. İşlemin son adımı, yoğunluk işlevinin üst sınır olarak X^2_{n-1} ile entegralini alarak gömme olasılığını, p, hesaplamaktır.

$$P = 1 - \frac{1}{2^{\frac{n-1}{2}}\tau(\frac{n-1}{2}} \int_0^{x^2_{n-1}} e^{-\frac{u}{2}} \; u^{\frac{n-1}{2}-1} \, du \tag{5.2}$$

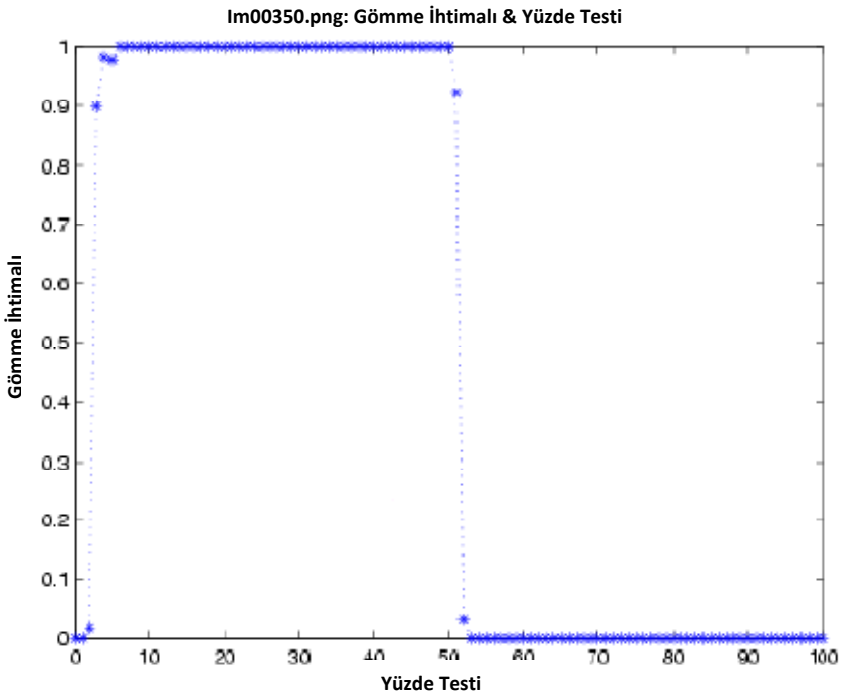
Bu gömme olasılığı Eşitlik (5.1)'deki bütün i'ler için $x_i = z_i$ koşulu altında X^2_{n-1} olasılığıdır. 1 - p yoğunluk fonksiyonu, X^2_{n-1} sonsuza yaklaşırken 1'e yakınsar, dolayısıyla X^2_{n-1} sonsuza yaklaşırken p 0'a yaklaşır. Dolayısıyla büyük X^2_{n-1} değeri için gömme olasılığı 0'a yakındır. Bununla birlikte, X^2_{n-1} n-1'a göre küçükken, 1 - p sıfırdır ve dolayısıyla p 1'e yakındır. Böylece görece küçük X^2_{n-1} değeri için gömme olasılığı 1'a yakındır. Ek olarak, Westfeld ve Pfitzmann eğer piksellerin % 100'ünden daha azı gömülü bilgi içeriyorsa, daha yüksek yüzdede pikseller test edildiğinde gömme olasılığı sert bir şekilde düşecektir.

5.4. Deneysel Sonuçlar

Test etmek üzere yedi adet değişik büyüklük ve düzgünlük ölçüsünde görüntü seçtim. Orijinal görüntüler "Im001", "Im002" ... "Im007" olarak etiketlenmiştir. Bu yedi görüntünün küçültülmüş sürümleri Ek A'da bulunmaktadır. Her birine, piksellerin % 10'u, % 50'si ve % 100 oranında bilgi gömdüm ve bu stego görüntünün her birini örneğin Im002 içerisindeki piksellerin % 10'una bilgi gömülmesi "FlipIm00210" adlı stego görüntü oluşturacak şekilde "Flip[görüntü adı][bilgi

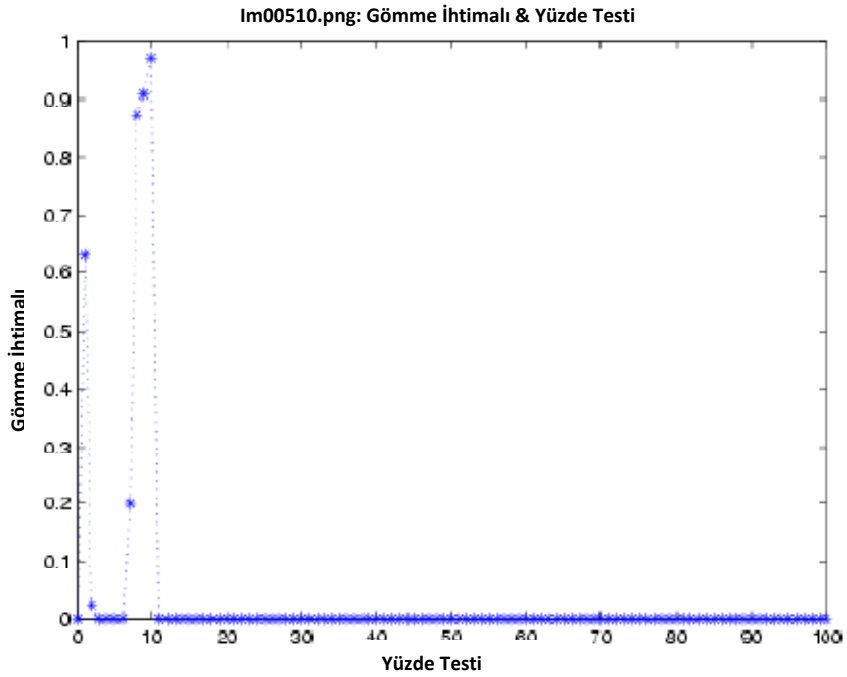
gömme yüzdesi]" formatında kaydettim. 21 stego görüntü oluşturduktan sonra bu stego görüntünün her birini ve orijinal görüntüyü (toplam 28 görüntü) DÇ3'ü kullanarak test ettim. Test edilen her bir görüntü için DÇ3 görüntüdeki piksellerin % 1, % 2 .... % 100'ü oranlarına karşılık gelen p olasılık Çizelgesunu geri döndürdü. Örneğin % 10'a karşılık gelen gömme olasılığı, test edilen görüntüdeki piksellerin ilk % 10'unun gömülü bilgi içerme olasılığıdır. Gösterim kolaylığı açısından, p_r test görüntüsündeki piksellerin ilk yüzde r'sinin gömülü veri içermesi olasılığını göstersin. Westfeld ve Pfitzmann'ın iddiasına göre, "Flip[görüntü adı][bilgi gömme yüzdesi]" görüntüleri test edildiğinde, p_r teorik olarak r > [gömme yüzdesi] için sert bir şekilde düşecektir

Seçtiğim görüntülerin örneklemesi küçük (yalnızca yedi belirgin görüntü) olmasına rağmen, bazı ilginç sonuçlar elde ettim. İlk olarak DÇ3, test ettiğim görüntülerin yalnızca yarısını özellikle büyük homojen bölgeleri olanları doğru bir şekilde analiz ediyor gibi görünmektedir. Örneğin Im003 bir kaç renge ve büyük düz bölgelere sahip olan ABD haritasının bir gri ölçekli bir görüntüsüdür. Bu görüntüde, DÇ3'ün neredeyse sıfıra yakın gömme olasılığı verdiği piksellerin ilk % 2'si haricinde DÇ3 bilginin gömüldüğü yerleri doğru bir şekilde algılamaktadır. Şekil 5.4 piksellerin % 50'sine bilgi gömülmüş olan Im003 olan FlipIm00350'nin verilerini göstermektedir. Gömülme olasılığının piksellerin % 51'i test edildikten sonra sert bir şekilde düştüğüne dikkat ediniz. Benzer düşmeler diğer görüntülerde de meydana gelmiştir. FlipIm00210'da, p_r, r > 11 olduğunda düşmekte ve FlipIm00250'de p_r, r > 52 olduğunda düşmektedir. Bununla birlikte Im005 ve Im004'teki veri gömme sonuçları daha az güvenilirdir. Im005'in piksellerinin % 10'una bitler gömüldüğünde Şekil 5.5 p_r $1 \leq r \leq 7$ i.in sıfıra yakın ve (tam olarak piksellerin % 10'u test edildiğinde meydana gelen değer olan 0,029'dan 1'e daha yakındır). Verilerime dayalı olarak, gömülü veriler içeren görüntü kısımları test edildiğinde virgülden sonra dört ya da beş haneye kadar testin doğru olması yaygın bir durumdur. Im005'deki piksellerin % 50'sine veri gömüldüğünde Şekil 5.6, p_r yine $1 \leq r \leq 7$ için sıfıra yakın ancak $8 \leq r \leq 50$ için bire çok yakın olup, bu noktadan sonra sert bir şekilde düşmektedir.

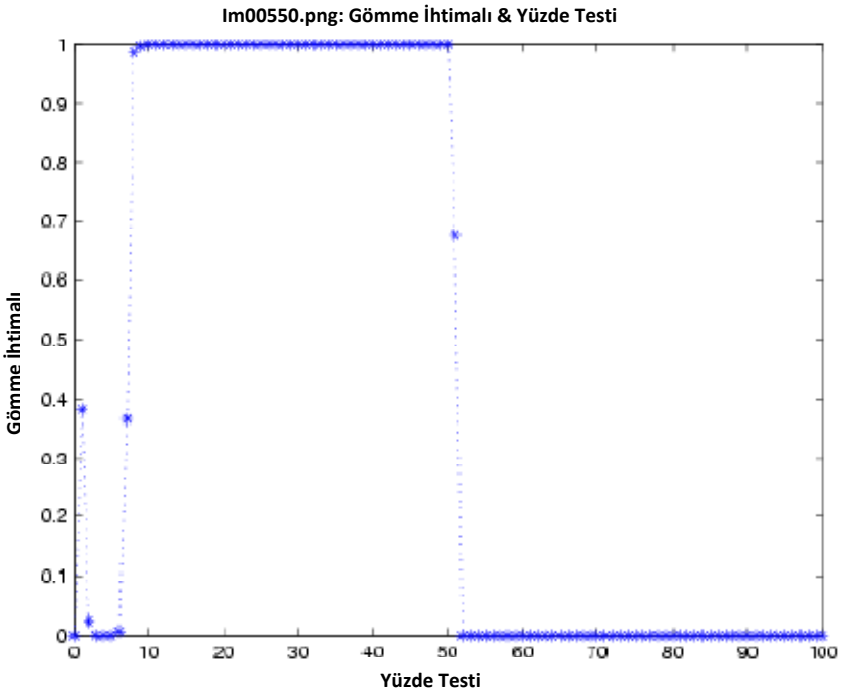


Şekil 5.4. FlipIm00350: Piksellerin % 50'sine veri gömülmüş Im003.

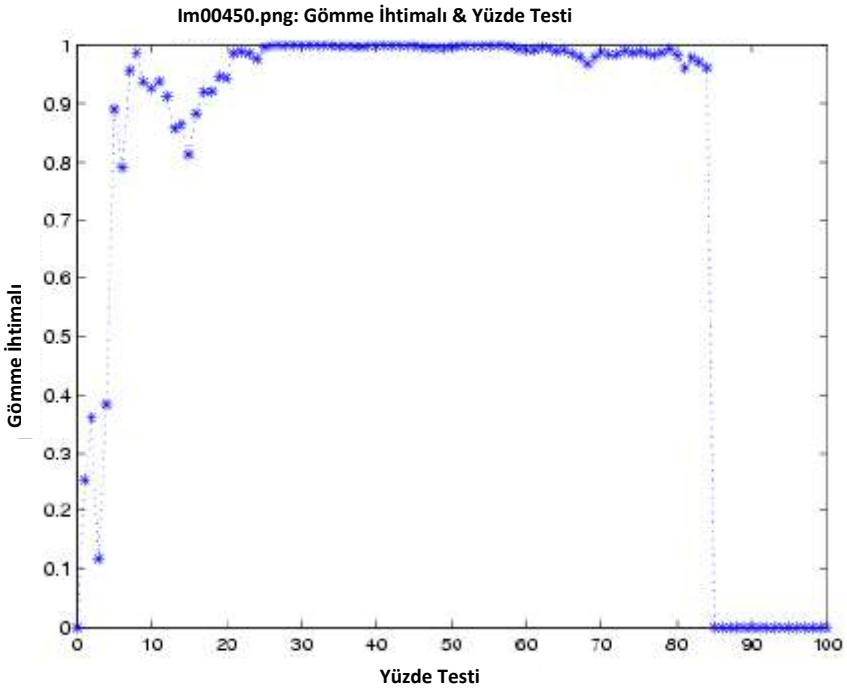
Im005'deki piksellerin % 50'sine veri gömüldüğünde Şekil 5.6, p_r yine $1 \leq r \leq 7$ için sıfıra yakın ancak $8 \leq r \leq 50$ için bire çok yakın olup, bu noktadan sonra sert bir şekilde düşmektedir. Gerçekte, p_r $11 \leq r \leq 50$ aralığı için virgülden sonra dört haneye kadar doğrudur. Im004'te farklı bir görüngü meydana gelmektedir. Im004'teki piksellerden % 10'u gömülü bitler içerdiğinde, düşme r = 12 iken meydana gelir ancak piksellerin % 50'sine bitler gömüldüğünde, düşme r = 85'e kadar meydana gelmemektedir Şekil 5.7. Dolayısıyla % 50'den % 85'e kadar olan pikseller bir stego görüntünde beklenenlere benzer özellikler sergilemektedir. Daha sonra % 50 ila % 85 arasındaki pikselleri (sıkı bir şekilde) test ettim ve bu bölgedeki gömme olasılığının 0,9685 olduğunu belirledim. Bu olasılık FlipIm00450'nin sonuçlarına uymakta ancak veri gömülmemiş bir görüntünün bölgelerinde teorik olarak beklenenin aksinedir.



Şekil 5.5. FlipIm00510: Piksellerin % 10'una veri gömülmüş Im005.



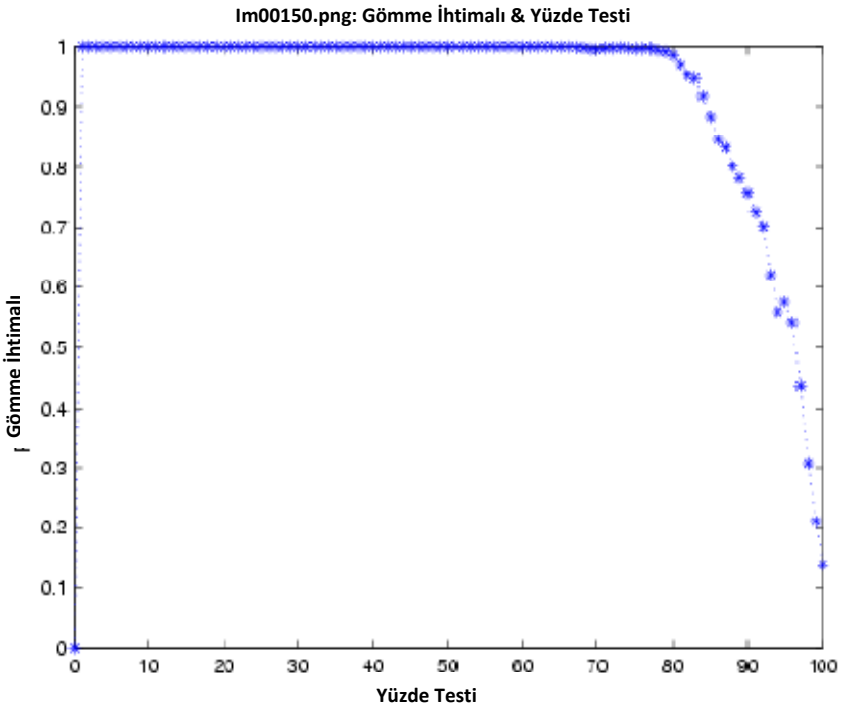
Şekil 5.6. FlipIm00550: Piksellerin % 50'sine veri gömülmüş Im005.



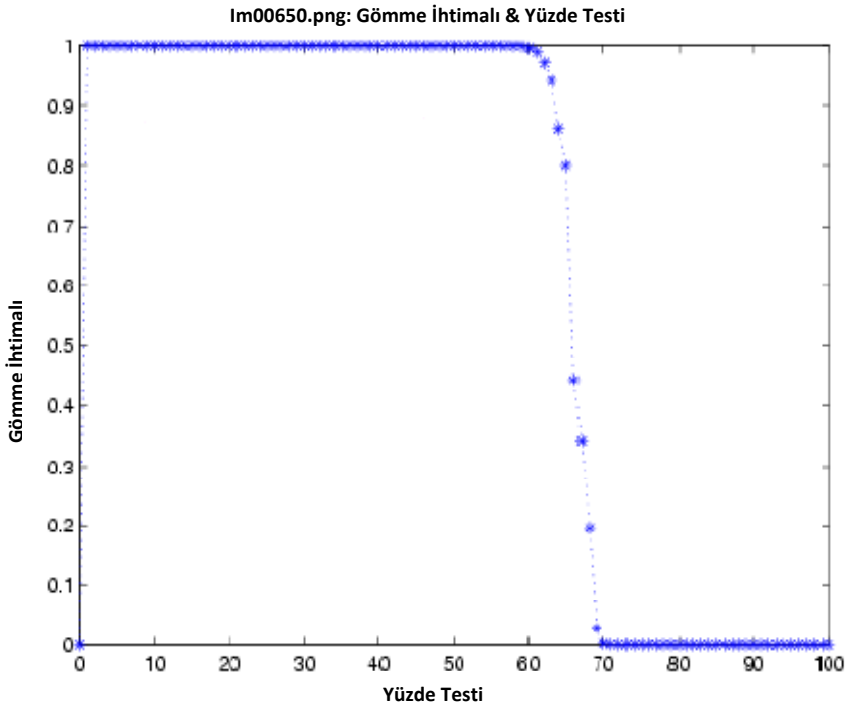
Şekil 5.7. FlipIm00450: Piksellerin % 50'sine veri gömülmüş Im004.

Bununla birlikte, gürültülü görüntüler test edildiğinde FlipIm00450'de gözlemlere benzer sonuçlar yaygın olduğu ortaya çıkmaktadır. Örneğin piksellerinin % 10'una bitler gömülmüş olan görüntüler olan FlipIm00110 ve FlipIm00610 olasılıkları p_r r = 18 olana kadar düşmemektedir. Piksellerin % 50'sine veri gömüldüğünde, FlipIm00150 Şekil 5.8 için r = 85 ve FlipIm00650 Şekil 5.9 için r = 64 olana kadar p_r düşmemektedir. Ancak Im004 için olan sonuçların aksine, Im001 ve FlipIm00150 için % 50 ve % 85 oranındaki piksellere veri gömme olasılıkları sıfıra çok yakındır. Bu durum Im006 ve FlipIm00650 için de geçerlidir.

Daha gürültülü olan görüntüler için sonuçlar daha az güvenilirdir. Örneğin, DÇ3 (veri gömülmemiş) Im007 ve (piksellerinin % 10'una veri gömülmüş Im007 olan) FlipIm007 için benzer sonuçlar vermektedir. Şekil 5.10 ve Şekil 5.11'deki eğrilerin benzerliğine dikkat ediniz. Gerçekte bu teste göre, Im007 gerçekte hiç bir gömülü veri içermemesine rağmen, piksellerinin % 30'undan fazlasına veri gömülmüş gibi görünmektedir. Bu sonuç Im007'nin ilk % 30'unun stego görüntülerde beklenenlere benzer özelliklere sahip olduğunu yani her bir değer çiftindeki her bir piksel değerinin sıklığının bu değer çifti için ortalamaya yakın olduğunu göstermektedir.

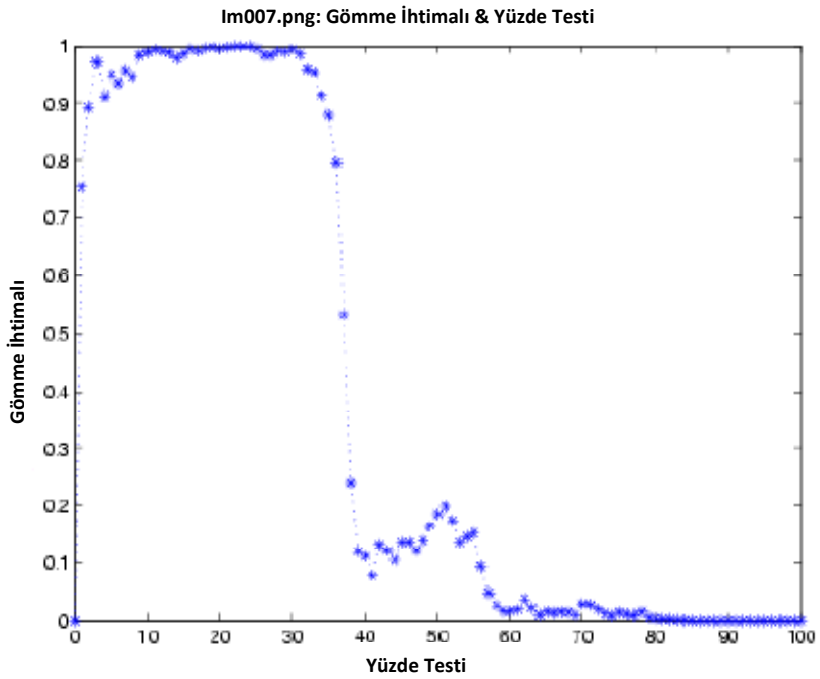


Şekil 5.8. FlipIm00150: Piksellerin % 50'sine veri gömülmüş Im001.

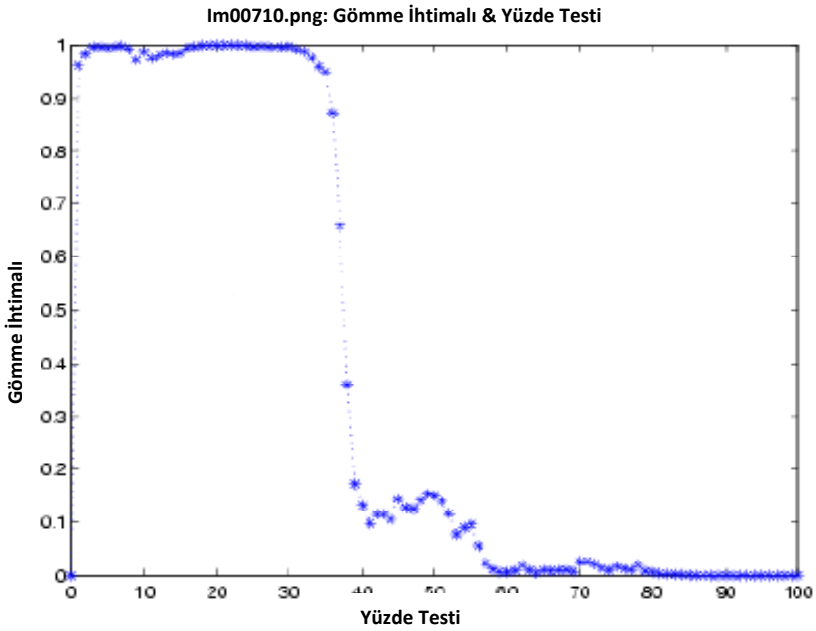


Şekil 5.9. FlipIm00650: Piksellerin % 50'sine veri gömülmüş Im006.

Bununla birlikte bu gibi bir sonuç şaşırtıcı olmayabilir. Önceki geleneksel görüşün en az önemli bitlerin tamamen rastgele oldukları yönünde olmasına rağmen, Westfeld ve Pfitzmann'ın iddiası görüntülerdeki en az önemli bitin tamamen rastgele olmadığıdır. Eğer bir görüntüdeki en az önemli bit tamamen rastgele ise, çift içerisindeki her bir piksel değerinin görüntü içerisinde ortaya çıkma olasılığının eşittir ve böylece yeterli veri verildiğinde her bir çiftteki her iki piksel değerinin sıklığı neredeyse eşit yani zk'ya yakın $0 \leq k \leq 127$'dir. Ki-kare saldırısı bir görüntü içerisinde her bir piksel çiftinin rastgeleliğini test etmek üzere tasarımlanmıştır, dolayısıyla rastgelelik özellikleri gösteren görüntüler için gömme olasılığının (yani rastgelelik olasılığının) bire yakın olması beklenir. Daha önce işaret edildiği gibi "gürültülü" olarak adlandırdığımız görüntüler özünde rastgele gibi görünmektedir. Böylece gürültülü görüntülerin bire yakın olasılığa sahip olması beklenebilir. Im004'te piksellerin ilk % 10'unun gökyüzünden (düz bir bölgeden) ve piksellerin son % 15'inin sokaktan (ki başka bir düz yüzeydir) oluştuğuna dikkat ediniz. Dolayısıyla görüntünün bu bölgelerinin rastgele olmama özellikleri gösterirken, görüntünün kalanının rastgeleliğe daha benzer olan özellikler sergilemesi mantıklıdır.



Şekil 5.10. Veri gömülmemiş durumdaki Im007.



Şekil 5.11. FlipIm00710: Piksellerin % 10'una veri gömülmüş Im007.

Bu durumda, sorulması gereken sonraki soru, görüntüdeki piksellerin % 100'ü test edildiğinde Im007'de veri gömme olasılığı neden bire yakın kalmadığıdır. Şekil 5.10. Görsel olarak görüntünün alt % 70'i üst % 30'u kadar rastgele görünmektedir. Ancak görüntünün görsel olmayan özellikleri piksellerin % 35'inden fazlası test edildiğinde veri gömme olasılığı sıfıra çok hızlı bir şekilde yaklaşacak şekilde değişim göstermektedir. Dahası Im001 ayrıca büyük oranda rastgele gözükmektedir, ancak DÇ3 Im001'i test edildiğinde pl yalnızca 1'in 0,04'ü içerisindedir. Diğer bütün r değerleri için p_r sıfıra yakındır. Bu daha fazla araştırma gerektirebilecek bir sorudur. Açık bir şekilde, yeterli rastgelelik gösteren örtü görüntüler seçilmesi Ki-kare testini özünde yararsız hale getirecektir.

5.4.1. Ki-kare saldırısının güvenilirliği

Yanıtlamaya başladığımız sorulardan birisi "Bir görüntünün gömülü bilgi içerip içermediğini söyleyebilir miyiz"'di ve basit yanıt "Bazen"'dir. Test edilen her bir görüntü bir görüntünün piksellerinin % 100'üne veri gömüldüğünde Ki-kare testinin bire çok yakın bir veri gömme olasılığı geri döndürmektedir. Diğer bir deyişle, (bu araştırmadaki) piksellerinin % 100'üne veri gömülmüş herhangi bir görüntü test

edildiğinde Ki-kare testi her bir görüntü için gerçek pozitif sonuç geri döndürmektedir. Çizelge 5.1'de piksellerin % 100'üne veri gömülmüş görüntüler için Ki-kare saldırısının sonuçlarını göstermektedir.

Çizelge 5.1. Flip[Görüntü]100: % 100 veri gömülmüş görüntüler.

İmaj % Test	İm001	İm002	İm003	İm004	İm005	İm006	İm007
% 85	EVET	EVET	EVET	EVET	EVET	EVET	EVET
% 90	EVET	EVET	EVET	EVET	EVET	EVET	EVET
% 95	EVET	EVET	EVET	EVET	EVET	EVET	EVET
% 100	EVET	EVET	EVET	EVET	EVET	EVET	EVET

Eğer $p_r \geq 0{,}90$ ise EVET, eğer $p_r \leq 0{,}10$ ise HAYIR, eğer $0{,}10 < P_r < 0{,}90$ ise BELİRSİZ.

Ki-kare saldırısı bu görüntüler için yalnızca gerçek pozitif sonuç vermesinden ve hiç bir yanlış değer vermemesinden dolayı, Ki-kare saldırısı piksellerin % 100'ü veri gömmek için kullanıldığında gizlenmiş verilerin varlığını algılamak üzere güvenilir bir şekilde uygulanabilir. Dahası piksellerin % 50'sine ya da daha azına veri gömülmüş görüntülerden hiç biri piksellerin % 100'ü test edildiğinde (sonuç verici bir şekilde) gömülü veriye sahip gibi görünmemektedir, dolayısıyla veri gömülmüş piksellerin tam yüzdesini belirleyemesek bile, örneğin piksellerin % 100'üne bilgi gömülmüş bir görüntü ile piksellerin % 50'sine ya da daha azına bilgi gömülmüş bir görüntü arasında ayrım yapabiliriz.

Ek olarak, Ki-kare testi çok az yanlış negatif sonucu geri döndürmektedir. Test piksellerin % 1 - 2'si test edildiğinde yalnızca FlipIm00310 ve piksellerin % 2- 6'sı test edildiğinde FlipIm00510 için yalnız negatif sonuç döndürmektedir. Verilerimde başka hiç bir yanlış negatif sonuç bulunmamaktadır. Yedi görüntünün her birindeki piksellerin % 10'una veri gömüldüğünde Ki-kare saldırısının sonuçları için Çizelge 5.2'de konulmuştur. Teoride yukarıdaki Çizelge 5.2'deki çift çizginin üzerindeki elemanların tamamı "EVET" olurken, altındaki elemanların tamamı "HAYIR" olmalıdır.

Çizelge 5.2. Flip[Görüntü]10: % 10 veri gömülmüş görüntüler.

İmaj %Test	İm001	İm002	İm003	İm004	İm005	İm006	İm007
% 1	EVET	BELİRSİZ	HAYIR	BELİRSİZ	BELİRSİZ	EVET	EVET
% 5	EVET	EVET	EVET	BELİRSİZ	HAYIR	EVET	EVET
% 10	EVET	EVET	EVET	EVET	EVET	EVET	EVET
% 15	EVET	BELİRSİZ	HAYIR	HAYIR	HAYIR	EVET	EVET
% 20	EVET	HAYIR	HAYIR	HAYIR	HAYIR	BELİRSİZ	EVET
% 25	BELİRSİZ	HAYIR	HAYIR	HAYIR	HAYIR	BELİRSİZ	EVET
% 30	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	EVET
% 35	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	EVET

Eğer $p_r \geq 0{,}90$ ise EVET, eğer $p_r \leq 0{,}10$ ise HAYIR, eğer $0{,}10 < P_r < 0{,}90$ ise BELİRSİZ.

Az önce söz edilen yanlış negatif sonuçlara ek olarak, üç görüntü de ayrıca yanlış pozitif sonuca sahiptir. FlipIm00110, FlipIm00610, ve FlipIm00710. Bununla birlikte, FlipIm00710 ve muhtemelen FlipIm00110 haricinde Ki-kare saldırısının sonuçlarına dayalı olarak mesaj uzunluğunu makul bir şekilde tahmin edebiliriz.

Genel olarak, piksellerin % 50'sine ya da % 100'üne (ve bu fikir test edilmemiş olmasına rağmen varsayımsal olarak % 50'den % 100'e kadar) veri gömülmüş durumdaysa, Ki-kare saldırısı gömülü veri olduğunu algılayacaktır. (% 50 veri gömülmüş görüntüler için Çizelge 5.3'de ve % 100 veri gömülmüş görüntüler için Çizelge 5.1'e başvurabilirsiniz.

Eğer veri piksellerin yalnızca % 10'una gömülüyse, bu durumda araştırmama dayalı olarak Ki-kare saldırısı piksellerin tam olarak % 10'u test edildiğinde veri gömülü olduğunu algılayacak ancak piksellerin % 10'undan azı test edildiğinde hiç bir gömülü veri olmadığını gösterecek ya da testin sonucu belirsiz olacaktır (Bakınız Çizelge 5.2).

Çizelge 5.3. Flip[Görüntü]50: % 50 veri gömülmüş görüntüler.

İmaj / %Test	İm001	İm002	İm003	İm004	İm005	İm006	İm007
% 35	EVET	EVET	EVET	EVET	EVET	EVET	EVET
% 40	EVET	EVET	EVET	EVET	EVET	EVET	EVET
% 45	EVET	EVET	EVET	EVET	EVET	EVET	EVET
% 50	EVET	EVET	EVET	EVET	EVET	EVET	EVET
% 55	EVET	HAYIR	HAYIR	EVET	HAYIR	EVET	EVET
% 60	EVET	HAYIR	HAYIR	EVET	HAYIR	EVET	EVET
% 65	EVET	HAYIR	HAYIR	EVET	HAYIR	BELİRSİZ	EVET
% 70	EVET	HAYIR	HAYIR	EVET	HAYIR	HAYIR	EVET

Eğer $p_r \geq 0{,}90$ ise EVET, eğer $p_r \leq 0{,}10$ ise HAYIR, eğer $0{,}10 < P_r < 0{,}90$ ise BELİRSİZ.

Bununla birlikte, Ki-kare testine göre bire yakın bir olasılık test edilen görüntünün gömülü bilgi içerdiğini her zaman ima etmemektedir. Örneğin test her iki görüntünün de herhangi bir gömülü veri içermemesine rağmen, Im001 için $p1 \geq 0{,}90$ ve Im007 için $p_r \geq 0{,}90$, $3 \leq r \leq 34$ sonuçlarını göstermektedir. (Bakınız Çizelge 5.4).

Çizelge 5.4. % 0 veri gömülmüş görüntüler.

İmaj / % Test	İm001	İm002	İm003	İm004	İm005	İm006	İm007
% 1	EVET	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	BELİRSİZ
% 5	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	EVET
% 10	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	EVET
% 15	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	EVET
% 20	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	EVET
% 25	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR	EVET

Eğer $p_r \geq 0{,}90$ ise EVET, eğer $p_r \leq 0{,}10$ ise HAYIR, eğer $0{,}10 < P_r < 0{,}90$ ise BELİRSİZ.

Bir görüntü içerisinde yeterli veri bulunduğunda, Ki-kare testi bunların varlığını algılayabilir ancak test gürültülü görüntüler ya da görüntülerin gürültülü olan bölgeleri için (FlipIm00450 için % 50 ve % 85 arasında olacak şekilde) yanlış pozitif değerler geri döndürebilir. Sonraki mantıksal soru "Bir görüntüye ne kadar veri gömüldüğünü söyleyebilir miyiz?"'dir. Bunu belirli bir dereceye kadar hâlihazırda yanıtladık. Westfeld ve Pfitzmann r p_r'nin düşmesine dikkat ederek gömme işleminin nerede durduğunu yüzde bir kaçlık bir hassasiyetle söyleyebildiklerini iddia etmektedir. Makalelerine göre [72], gömülmüş veriden sonraki pikseller test edildiğinde p_r sert bir şekilde düşmektedir. Bununla birlikte test etmiş olduğum çoğu görüntü için bu durumun doğru olduğunu tespit etmedim. Dolayısıyla yedi görüntü için düşmelerin nerede meydana geldiğini gösteren çizelge 5.5, hem p_r'nin 0,90'nın altına düştüğü hem de p_r'nin altına düştüğü yerleri göstermektedir.

Çizelge 5.5. Ki-kare saldırısına göre "Düşmeler".

İmaj % Emb.	İm001	İm002	İm003	İm004	İm005	İm006	İm007
% 0	1/5	0/0	0/0	0/0	0/0	0/0	34/41
% 10	20/26	11/17	10/12	12/15	10/11	17/30	35/41
% 50	84/100+	52/54	51/52	84/85	50/52	63/69	82/100+

$p_r \geq 0{,}90$ olacak şekilde test edilen piksellerin en yüksek yüzdesi /
$p_r < 0{,}10$ olacak şekilde test edilen piksellerin takip eden yüzdesi

Daha önce dikkat çekildiği gibi, Ki-kare testi daima görüntü içerisine ne kadar veri gömülü olduğunu yüksek doğrulukla göstermemektedir. Hem FlipIm00150 hem de FlipIm00450, yalnızca piksellerinin % 50'sinde veri içermelerine rağmen Westfeld ve Pfitzmann'ın iddiasına göre piksellerinin % 85'ünde veri içermektedir. Benzer bir şekilde FlipIm00750 piksellerin % 82'si test edilene kadar düşmemekte ve FlipIm00650 piksellerin % 63'ü test edilene kadar düşmemektedir. Ki-kare testi piksellerin % 50'si veri içerdiğinde yedi görüntünün yalnızca üçünde ve piksellerin % 10'u veri içerdiğinde yedi görüntünün yalnızca dördünde ne kadar veri gömülü olduğunu doğru bir şekilde göstermektedir.

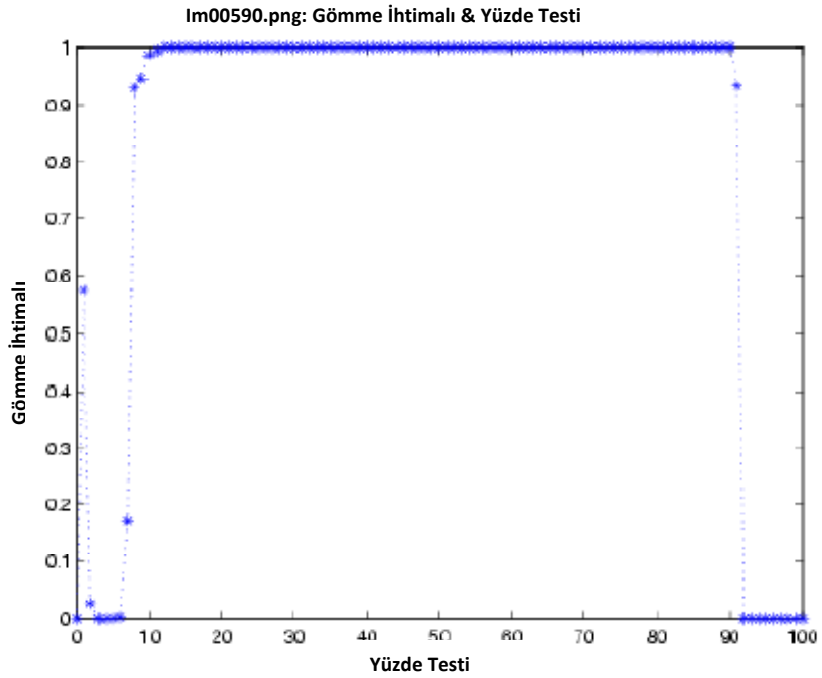
5.4.2. Ki-kare saldırısının yenilmesi

Ki-kare saldırısının açık zayıflıkları, Ki-kare saldırısını etkisiz ya da güvenilmez yapacak şekilde veri gömme fırsatları sunuyor gibi görünmektedir. DÇ3'ü şu anda yazıldığı şekilde bertaraf edebilecek bu fırsatları tartışacağım. DÇ3 algoritması bu suiistimalleri yenmek üzere bazı durumlarda değiştirilebilir.

Görüntü seçimi sıklıkla Ki-kare saldırısının zayıflıklarının bazılarından yararlanan görüntüler seçebiliriz. Bu yöntemlerden birisi, görüntünün mümkün olduğu kadar tamamında istatistiksel olarak gürültülü olan görüntüler seçmektir. Hâlihazırda Ki-kare saldırısının pek çok gürültülü görüntüler için güvenilmez olduğunu hâlihazırda gösterdik. Dahası, gömülü veri içeren ve içermeyen görüntünün istatistiksel olarak aynı görünmesi için görüntünün toplam piksellerinin belirli bir yüzdesine kadar bazı görüntülere bilginin gömülebileceğini gösterdik. Gürültülü görüntülerin steganografi için en iyi görüntüler olduğu oldukça iyi belgelenmiştir.

Bununla birlikte örtü görüntüye ek kısıtlamalar dayatarak ve gömme algoritmasını değiştirerek görüntü seçeneğini iyileştirebiliriz. Örneğin görüntünün üst kısmında büyük homojen bir bölge ve görüntünün kalanı boyunca gürültüye sahip bir görüntü seçebiliriz (DÇ3'ün test ederken görüntünün üst kısmında başladığını unutmayınız) ve gürültülü bölgeden veri gömmeye başlayabiliriz yani üst kısımdaki homojen bölgeyi atlayabiliriz. Bu gibi algoritma steganografik sistemin kapasitesini azaltacak ancak görüntünün üst kısmında büyük bir homojen bölge olması tüm görüntü için gömme olasılığını sıfıra yakın kalmasına neden olmalıdır. Gerçekte bu gibi bir gömme algoritması kullanılmış ve görüntüler yalnızca üstten alta doğru test edilmiş ve daha sonra tüm görüntü düzgün olabilmiş ancak Ki-kare saldırısı hala görüntüye hiç bir veri gömülmemiş olduğunu henüz gösterememiştir. Örneğin piksellerin % 90'ına bit gömülmüş FlipIm00590, Im005 görüntülerini düşünüz Şekil 5.12. Eğer bu görüntü, veri gömülmemiş alan üstte olacak şekilde 180 derece döndürülürse ve DÇ3 kullanılarak test edilirse, DÇ3 piksellerin % 100'ü test edildiğinde gömülmüş veri olmadığını gösterecektir. Bu durum FlipIm00590'daki piksellerin % 100 görüntüyü 180 derece çevirmeden test edildiğinde veri gömme olasılığının sıfıra çok yakın

olmasından aşikârdır. Dolayısıyla döndürülmüş FlipIm00590 DÇ3 kullanılarak test edildiğinde, algoritma bütün r için p_r sıfıra yakın döndürecektir. Diğer bir deyişle, test işlemi düzgün bir bölgeden başladığı sürece görüntünün ne kadarının test edildiğine bakılmaksızın DÇ3'ün görüntüde gömülü veri olmadığını göstermesini bekleyebiliriz. Gerçekte FlipIm0590'ı çevirdim ve DÇ3'ü kullanarak test ederek böylece (bu görüntü için) bu durumu teyit ettim. Bununla birlikte bu görüntüyü DÇ2r kullanarak test edersek, gömme olasılığının neredeyse bire yakın olduğu % 10'dan % 100'e kadar olan oranlarda pikselleri test edebiliriz. Gömme olasılığının veri gömülü olduğunda, yalnızca düz görüntüler için bire yakın olmasından dolayı, bu görüntünün daha sonraki kısmının gerçekte gömülü veri içerdiğini belirleyebiliriz.

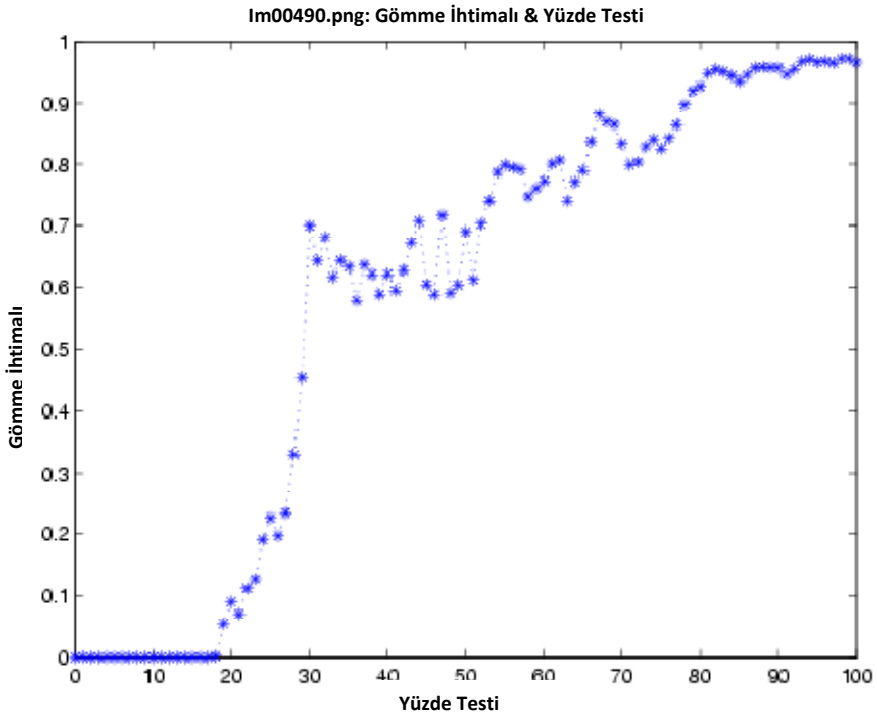


Şekil 5.12. FlipIm00590: Piksellerin % 90'ına veri gömülmüş Im005.

Ayrıca Im004'le benzer bir deney de gerçekleştirdim, yani piksellerin ilk % 90'ına veri gömdüm ve görüntüyü 180 derece döndürdüm. DÇ2r'e göre, Im004 içerisindeki piksellerin son % 10'u istatistiksel olarak çok düzgün görünecektir. Bununla birlikte 180 derece döndürülmüş FlipIm00490 DÇ3 kullanılarak test edildiğinde, Ki-kare testi beklediğim gibi görüntüde hiç bir veri gömülü olmadığını göstermektedir. Daha çok test görüntünün ilk % 21'inin hiç bir gömülü veri içermediğini, görüntünün son

% 21'inin gömülü veri içerdiğini ve testin görüntünün kalan % 54'ü için belirsiz olduğunu göstermektedir. 180FlipIm00590 (180 derece döndürülmüş FlipIm00590) sonuçları ile 180FlipIm00490 (180 derece döndürülmüş FlipIm00490) sonuçları arasındaki fark önceki görüntüdeki düz bölgenin sonraki görüntüdekinden daha homojen olmasının bir sonucu olabilir. Eğer $|x_k - z_k|$ bir kaç değer çifti için bile çok büyükse bu durumda, X^2_{n-1}'in büyük olduğu dolayısıyla veri gömme olasılığının sıfıra yakın olduğu makul bir şekilde umulabilir. Eğer 180FlipIm00590'ın düz bölgesi yalnızca örneğin 4 değer çiftine karşılık gelen piksel değerleri içeriyorsa ancak 180FlipIm00490'nın düz bölgesi diyelim ki yalnızca 10 değer çiftine karşılık gelen piksel değerleri içeriyorsa, bu durumda düzgün bölgelerde meydana gelen sırasıyla 4 ya da 10 kategori için $|x_k - z_k|$ 180FlipIm00590'da 180FlipIm00490'da olduğundan daha büyük olabilir. Eğer bölge düzgünse, hipoteze göre $|x_k - z_k|$'nın büyük olduğunu unutmayınız. Böylece eğer 10 kategori için |xj -zj|'nin toplamı i, 180FlipIm00490'da, 180FlipIm00590'da 4 kategori için |xj -zj|'nin toplamı j'den küçükse, görüntünün kalanı (gömülü veri olan kısmı) 180FlipIm00490 için n - 1'e göre X2n-1 büyüklüğünü, 180FlipIm00590 için n - 1'e göre X2n-1 büyüklüğünden daha kolay yenebilir.

Veri gömmek için piksellerin seçilmesi. Eğer $|x_k - z_k|$ bir kaç değer çifti için bile çok büyükse bu durumda, X^2_{n-1}'in büyük olduğu dolayısıyla veri gömme olasılığının sıfıra yakın olduğu fikri diğer gömme algoritmalarına genişletilebilir. Gömme algoritmasını değiştirmenin bir yolu verinin hangi piksellere gömüleceğini seçmektir. En azından $|x_k - z_k| > \tau$ olacak şekilde u adet değer çifti olduğunda veri gömmenin sıfıra yakın kaldığını varsayınız. $|x_k - z_k|$'nin τ'ye göreceli olarak büyük olduğu en azından u adet değer çiftinin bulunduğu bir örtü görüntü seçiniz. Daha sonra mesajı yalnızca $|x_k - z_k| > \tau$ iken 2k ve 2k+1 piksel değerlerine gömünüz. $|x_k - z_k| = \tau + 1$ iken, 2k ve 2k + 1 piksel değerleri artık gömme için kullanılamaz. Böylece eğer veri gömme olasılığı gerçekten en azından $|x_k - z_k| > \tau$ olacak şekilde u adet değer çifti varken sıfıra yakın kalıyorsa, görüntü istatistiksel olarak gömülü veri yok gibi görünecektir.

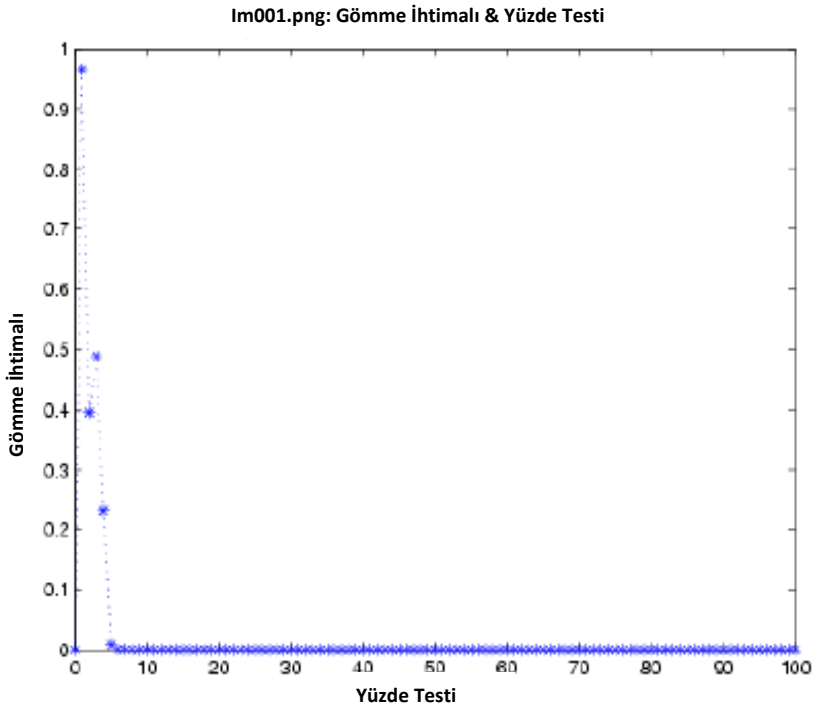


Şekil 5.13. 180FlipIm00490: Piksellerin ilk % 90'ına veri gömülmüş ve daha sonra 180° döndürülmüş Im004.

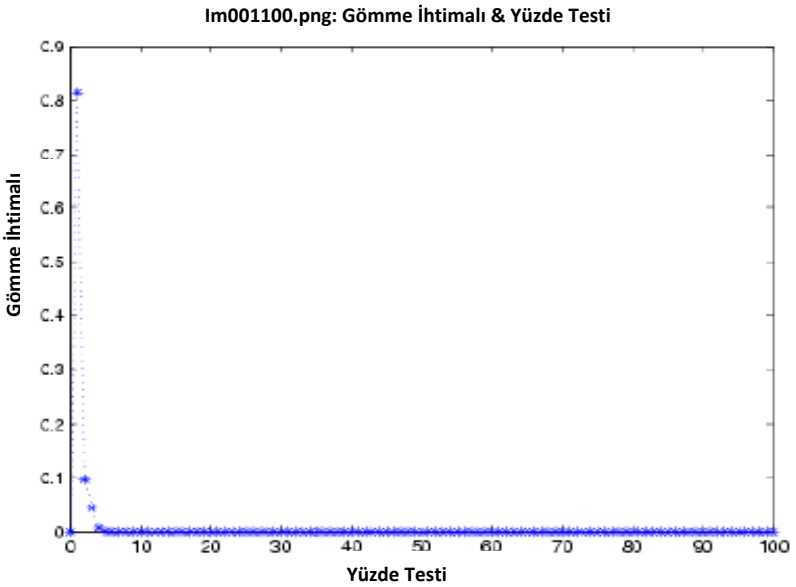
Dahası, bu piksellere de veri gömerek $x_k + y_k \leq 4$, $x_k = y_k = z_k = 0$ ve n = n-1 olduğunu belirten minimum sıklık koşulundan da yararlanabiliriz. Piksel değerlerinin sıklık sayımlarının Ki-kare testine uymaması nedeniyle, bu piksellere veri gömülmesi veri gömme olasılığını etkilemeyecektir. İdeal olarak bu algoritmada mesajı almak için, basitçe $|x_k - z_k| > \tau$ ya da $x_k+y_k \leq 4$ için 2k ve 2k + 1 piksel değerlerinin en az önemli bitini okuyabiliriz. Bununla birlikte, görüntüdeki son çiftin piksel değerine ulaşmadan önce belirli piksel değeri kullanılmaz hale gelirse, mesajın alınması zor olabilir. Ne var l, bu fikir daha fazla araştırma (ve yaratıcılık) gerektirebilir.

Aynı prensibe dayalı diğer bir fikir de yalnızca bir değer çiftindeki her ikinci ya da üçüncü piksele veri gömmek ve $|x_k - z_k|$'nin τ toleransından daha büyük olmaya zorlamak üzere her bir çiftteki kalan piksel değerlerini değiştirmektir. Böylece hangi piksellerden mesajı çıkartılacağı bilinebilir ancak Ki-kare testini başarısız kılmak üzere bütün k değerleri için $|x_k - z_k|$ yeterince büyük hale getiremeyebilir.

Değer çiftlerinden kaçınılması, Ki-kare saldırısının (Westfeld ve Pfitzmann tarafından teyit edilen) belirgin bir zayıflığı da yalnızca en az önemli biti değiştiren ve böylece değer çiftleri oluşturan veri gömme algoritmaları üzerinde etkili olmasıdır. Gömmek2 en az önemli biti değiştirmek yerine stego görüntüyü elde etmek üzere piksel değerini 20 azaltan benim yazdığım ikinci bir algoritmadır. Mesajın alınması FE aracıyla aynıdır: M = mod(S, 2). Bu algoritma kullanılarak piksellerin % 100'üne bilgi gömülmüş görüntüler için, p_r görüntüye hiç bir bilgi gömülmediğinde ortaya çıkan değerine yakın olma eğilimindedir. Örneğin, Im001 Şekil 5.14 ve Gömmek2 kullanılarak piksellerin % 100'üne veri gömülmüş DecIm001100'ün Şekil 5.15 p_r grafiklerini karşılaştırınız. DecIm1100'ün sonuçları, gerçekte piksellerinin tamamına veri gömülmüş bir görüntüde veri gömülme olasılığının daha az olduğunu göstermektedir. Benzer bir şekilde, Gömmek2 kullanılarak piksellerin % 100'üne veri gömülen yedi görüntünün her biri için faydasız sonuçlar oluşturulmuştur.



Şekil 5.14. Veri gömülmemiş durumdaki Im001.



Şekil 5.15. DecIm001100: (Gömmek2 kullanılarak) piksellerin % 100'üne veri gömülmüş Im001.

6. SES TABANLI STEGANALİZ UYGULAMALARI

Bu bölümde steganografik araçlarda FE aracı kullanarak Wavelet (WAV) ve Audio Unit (AU) dosyasında saklı mesajın belirlenmesi için steganalitik yaklaşımları sunulmuştur. Deney sonuçları steganografik kapasitenin %10'i kadar küçük, yerleştirilen mesajların güvenilir bir şekilde belirlenebildiğini göstermektedir.

Dijital içeriğin ve geniş internet iletişim sisteminin baş döndürücü hızlı gelişimi ile steganografi gittikçe daha fazla popüler hale gelmektedir. Yüksek tanım dijital görüntü ve doğala en yakın ses dijital fotoğraf makinesi ve dijital ses kayıt cihazı kullanarak uygun bir biçimde elde edilebilir. Görüntüde seste ve video dosyalarında saklayabilen birçok steganografik araç artık internette mevcut durumdadır. İnsanlar bu tür aletleri kullanarak gizli veriyi bu dijital içeriklere saklayabilirler ve gönderebilirler ya da gizlilikleri sağlayabilmek için disklerinde tutarlar. Sadece hedef alıcılar ya da kendileri bu gizli veriyi alabilirler. Üçüncü şahısların gizli veriden ya da saklı iletişimden haberleri olmaz. Yani bu bazı durumlarda kriptografiden çok daha güvenlidir.

Ancak bu iki uçlu bir kılıç gibidir ve yanlış kullanılabilir. Ses dosyalarında veri saklama metotları başlıca düşük-bit şifreleme (Low-Bit Encoding), Aşama kodlama (Phase Encoding), Taft yayılması (Spread Spectrum) ve Yankı veri gizleme (Echo Data Hiding) diye ayrılır [13,14,15]. Düşük-bit şifreleme (Low-Bit Encoding) metodu çift akım aracılığıyla örnek verinin son bitinin yerine gizli olanı kullanmaktır. Bu en basit yöntemdir ve yüksek kapasite sağlar. Ama lossy sıkıştırmadan (lossy compression) ya da format dönüşümünden (format conversion) kaynaklanan küçük değişikliklere karşı koruma sağlayamaz. Faz şifreleme metodu (Phase Encoding), ilk ses dilim safhasını veriyi temsil eden referans bir safhayla değiştirerek çalışır. Yayma spektrumu (Spread Spectrum) veriyi orijinal sinyalin frekans spektrumunda saptırma tekniğidir. Yansıma veri saklaması (Echo Data Hiding) bir eko ortaya koyarak datayı ev sahibi bir sinyale yerleştirmektir. Verile ekonun üç parametresini değiştirerek yerleştirilir: ilk genişlik, bozulma oranı ve bedel. Steganografiye resmi karşı önlem olan steganaliz, verilen ortam içerisindeki

gizli bir bilgiyi saptamak ve genel olarak bozmaktır. Dijital bir görüntünün steganalizi ile karşılaştırıldığında dijital bir sesin steganalizi nispeten daha az keşfedilir. Sesteki gizli bir bilginin varlığının/yokluğunun insan kulağıyla saptanması mümkün değildir.

Yerleştirme işlemi genel olarak taşıyıcının istatistiklerini değiştirir. Bu sebepten dolayı birçok steganalitik teknik istatistikî modele dayandırılmaktadır [16,17].

Resim steganalizi konusundaki kadar çok olmasa da ses steganalizi konusunda da bazı çalışmalar yapılmıştır [71,41,59].

Ru ve ark. (2006), mevcut bazı steganografi yazılımlarıyla WAV dosyalan İçerisine yapılan gömme işlemini tespit etmeye yönelik bir yöntem geliştirmişlerdir [71].

Özer ve ark. (2006), saklama algoritmasının bilinmesine gerek duyulmaksızın, ses dosyaları içerisinde saklı mesajın varlığını evrensel bir şekilde tespit etmeye yönelik yaptıkları çalışmada %75 ile %90 arasında değişen bir başarı etmişlerdir [41].

Özer ve ark. ses kalitesi ölçülerine kurulu bir metot ortaya koymuştur [18]. Temel fikirleri, Örtü ses sinyalleri ve stego-ses sinyalleri üzerinde hesaplanan çeşitli istatistikî uzaklık ölçülerinin dağıtımı ve bire bir sesli sürümlerinin istatistikî olarak farklı olduğu yönündedir. Steganalizer olarak uygun bir ses kalitesi ölçeri seçtiler ve iki sınıftan meydana gelmekte olan bir sınıflandırıcı yaptılar.

Micah Johnson ve ark. [19], Yenilerde ses sinyallerinin istatistikî düzenlemelerini alabilmek için bir program sundular ve sınıflandırma için de doğrusal olmayan bir destek vektörü kullandılar. Ancak, bu yaklaşımın düşük-bit oranlı yerleştirmeleri saptamakta etkili olacağı muhtemel değildir. Steghide, [20] mesajları JPEG, BMP, WAV, ya da AU dosyalarına yerleştirebilen internet üzerinde ücretsiz olarak bulunabilen bir araçtır. Birincil emir istatistiksel saldırılara karşı koyabilmek için

tasarımlanmıştır.

Ses steganalizi konusundaki yaklaşımları ise aşağıdaki şekilde listeleyebiliriz[92]:
- İşitsel tespit
- Algoritmaya özel tespitler ve
- Evrensel tespit.

İşitsel tespit sestekine benzer şekilde stego ses dosyasının ses kalitesi yönüyle orijinalinden ayırt edilmesini veya orijinal dosya elde yoksa normal bir seste olmaması gereken bozuklukların tespitini ifade eder. Ancak kendinden gürültülü seslerle stego sesleri ayırt etmek yine kolay olmayacaktır.

Algoritmaya özel tespitler belirli algoritmalarla yapılan saklama işlemlerinin varlığını tahmin etmeye çalışan yöntemlerdir.

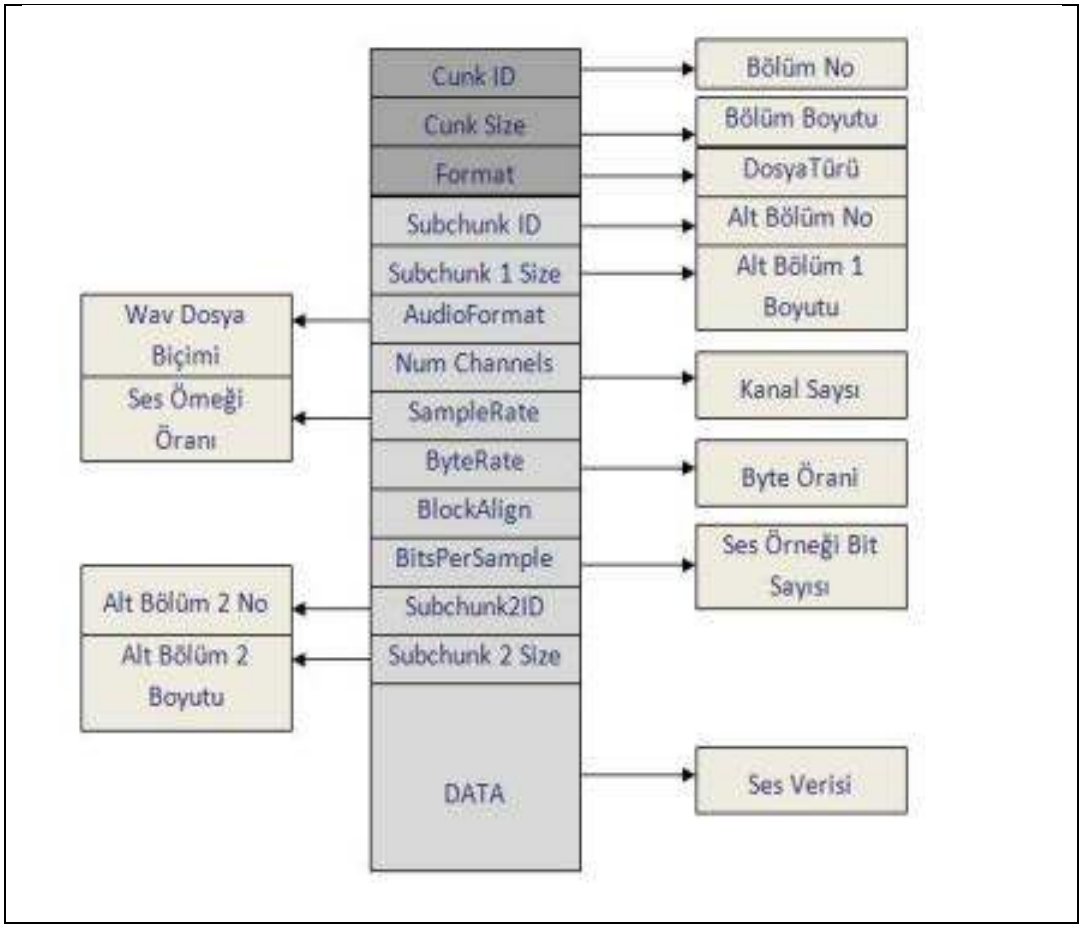
Evrensel tespit ise orijinal ses dosyalan ile stego ses dosyaları arasındaki çeşitli özelliklere göre farklılıkları tespit edip taşıyıcı dosya ile stego ses dosyası ayrıştırmaya yönelik yöntemleri içermektedir.

6.1. WAV Dosya Yapısı

Yüksek kaliteli ses örneklerini temsil eden en popüler format, 16 bitlik Windows Audio-Visual (WAV) dosya formatıdır [65]. Bir WAV dosyası, diğer dosya türlerinde de olduğu gibi bir başlık bölümüyle, bunu takip eden ve içerisinde iki adet alt veri parçası barındıran bir "WAVE" veri parçasından oluşmaktadır. Bu veri parçasının içindeki alt veri parçalarından birincisi "fmt" veri parçası olup, WAV dosyasının veri biçimiyle ilgili bilgileri içermektedir. Diğer alt veri parçası olan "data" ise asıl ses örnekleri verisini içermektedir.

WAV dosyasının veri bölümü ses örneklerinden (Audio Samples) oluşmaktadır. Bir WAV dosyasının ses örnekleri 8 veya 16 bit uzunluğunda olmaktadır [66]. Yani bir ses örneği bir veya iki byte olabilmektedir. Ses örnekleri, doğası gereği, bir ses

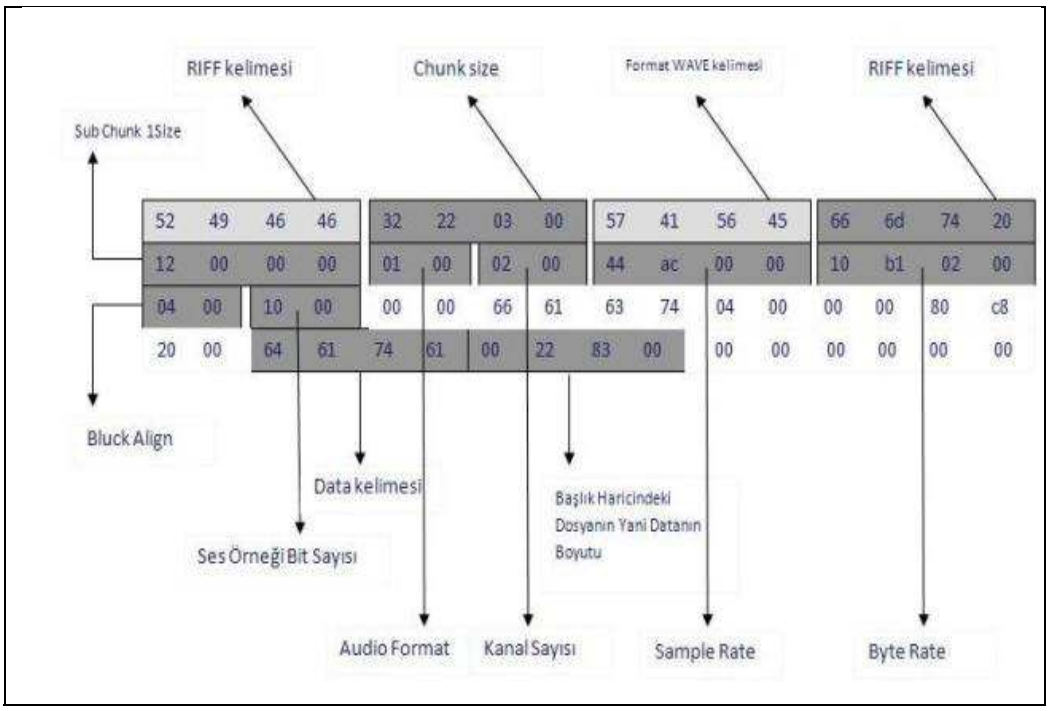
dalgasının belirli bir andaki kesin olmayan yaklaşık değerini ifade etmektedir [67]. Bu durumun bize, bu ses örneklerinin en az önemli bitleriyle oynama imkânı verdiğini belirtebiliriz. Şekil 6.1'da WAV dosya formatı gösterilmişti. Bu şekilde gösterilen veri alanlarından uygulama açısından önemli olanları açıklayalım *ChunkID* kısmında "RIFF" kelimsi yazılıdır. RIFF WAV dosyalarını da içeren genel dosya formatının ismidir [23].



Şekil 6.1. WAV dosyanın formatı

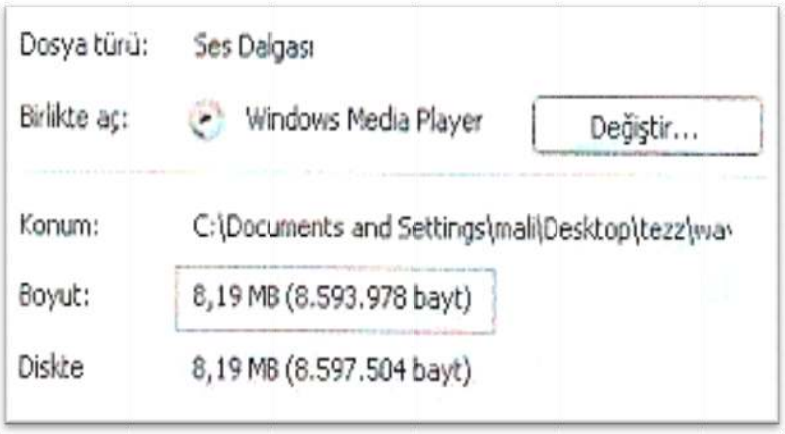
ChunkSize alanındaki değer, kendisinden sonra gelen tüm bölümlerin toplam boyutunu byte cinsinden belirtmektedir. *Format* kısmında "WAVE" kelimesi yazılıdır. *Subchunk1ID* kısmında "fmt" yazılıdır ve format ile ilgili bilgiler başlamaktadır. *Subchunk1Size* değeri ise, kendisiyle *Subchunk2ID* alanı arasında kalan bölümlerin toplam boyutunun ne kader byte olduğunu göstermektedir. *AudioFormat* WAV dosyasının biçimini gösterir. *PCM* için bu değer 1 olmalıdır.

Bunun dışındaki değerler sıkıştırma olduğu anlamına gelmektedir. *NumChannels* değeri kaç kanal olduğunu gösterir. *Mono* için 1, *Stereo* için 2 değerini alır. *BitsPerSample* verisi bir ses örneğinin kaç bitlik bir değere sahip olduğunu göstermektedir. WAV dosyaları için bu değerin 8 veya 16 olduğu yukarıda belirtilmişti. *Subchunk2ID* kısmında "data" yazmaktadır. *Subchunk2Size* kısmında başlık kısmının haricindeki asıl datanın büyüklüğünü vermektedir Şekil 6.2. WAVE formatındaki bir ses dosyasının başlık kısmındaki veriyi göstermektedir.



Şekil 6.2. Örnek WAV dosya başlığı

Şekil 6.2'de görüldüğü gibi, WAV dosyası en başta ASCII formatında RIFF kelimesi içermektedir. Daha sonraki bölümde, yani ChunkSize değeri, kendisinden sonra arta kalan tüm dosyanın boyutunu göstermektedir. Yani tüm dosya boyutundan, RIFF için ve kendisi için ayrılan 8 byte uzunluğundaki kısım haricindeki boyutu ifade eder. Yukarıdaki örnek dosya için ChunkSize bölümündeki değer: 256^0 (1*2+16*3) +256^1 (1*2+16*2)+256^2(1*3+16*8)+256^3(1*0+16*0)=8593970 Byte büyüklüğündedir. Buna 8 byte değerini elde ederiz. Ses dosyasının özellikler ekranına Şekil 6.3 bakarak bunu doğrulayabiliriz.



Şekil 6.3. WAV dosya özellikler ekranı

Şekil 6.3'den görüldüğü gibi dosyanın boyutu, başlık kısmındaki ChunkSize değerinden yola çıkarak hesapladığımız değeri göstermektedir. Bunun haricinde ses örneği (audio sample) bit sayısı bölümündeki değer:

$256^{0}*(1*0+16*1)+256^{1}*(1*0+16*0)=16$

Yani örnek WAV dosyası 2 byte büyüklüğündeki oluşmaktadır. Bu değerin bilinmesi saklama işleminde önemlidir. Saklama işlemi yaparken, taşıyıcı ses dosyasının başlık kısmındaki tüm bölümler okunmakta ve herhangi bir problem olduğu anlaşıldığında saklama işlemi iptal edilmektedir [23].

Resimlerle benzer olarak, veri saklama yöntemleri insanın işitme sistemini istismar ederek çalışırlar. Ses dosyalarındaki veride oluşacak küçük değişimler insan kulağı tarafından ayırt edilemez. Veri saklama yöntemi insanın işitme sisteminin (Human Audotary System - HAS) resimleri yorumlama yöntemini istismar eder.

HAS belli bir seviyenin altındaki sesleri ihmal ederken belli bir seviyenin üstündeki seslere odaklanmaktadır. Steganografik yöntemlerin amacı bu özelliği kullanarak veri saklamaktır. İnsan kulağının işitme sınırların dışındaki sesi oluşturan bitler veri saklamak amacıyla kodlanabilmektedir [8].

Ses dosyalarına veri saklamak için de farklı yöntemler kullanılmaktadır. Resimlerde

olduğu gibi ses içine veri saklamada da en sık kullanılan yöntem LSB yöntemidir. Burada ise piksel değerlerine karşılık olarak ses örnekleri (audio samples) veri saklama amacıyla değiştirilmektedir [8].

Mp3 dosyaları ve WAV dosyaları en sık kullanılan dijital ses dosyalarıdır. WAV dosyaları temel olarak iki kısımdan oluşmaktadır. Bunlar başlık kısmı ve veri kısmıdır. Başlık kısmı WAV dosyasına ve veri kısmına ait çeşitli bilgilerin tutulduğu kısımdır. Veri kısmında ne kadar byte'lık veri olduğu bir veri örneğinin kaç byte'lık değerden oluştuğu gibi, veri saklamada dikkate alınacak bazı parametreler başlık kısmında tutulmaktadır. Geri kısmı ise ses örneklerin yer aldığı kısımdır. LSB yöntemiyle WAV dosyasının içine veri saklanacağı zaman bu işlem için WAV dosyasının veri kısmı kullanılır.

Başlık kısmında yapılacak en küçük bir değişiklik dosyanın formatına zarar verebileceği için bu kısma hiç dokunulmaz. Zaten başlık kısmı veri kısmının yanında boyut olarak ihmal edilebilir bir değere sahiptir.

Resimdekinden farklı olarak WAV dosyasına veri saklanırken ses örneklerini oluşturan bütün byte değerlerinin en az önemli bitleri yerine en az önemli byte değerinin en az önemli bit değeri saklamada kullanılır [8].

Hatırlanacağı gibi resimde bir piksel eğer 3 byte değerinden oluşuyorsa bütün byte değerlerinin son bitleri veri saklama amacıyla değiştirilebiliyordu. Ancak WAV dosyasında eğer bir ses örneği 2 byte değerinden oluşuyorsa bu 2 byte değerinin de son bitlerinin değişmesi seste bozukluğa neden olabilmektedir. Bu nedenle veri saklarken en az önemli byte değerinin bitleri kullanılmaktadır.

Wav dosyalarında ses örnekleri 8 veya 16 bitlik değerlerden oluşmaktadır. Yani bir ses örneği ya bir yada iki byte'lık değerlere sahip olabilmektedir. Ses örnekleri bir ses dalgasının belli bir andaki tahmini değerinden oluşmaktadır. Buradaki esneklik veri saklama için kullanılabilmektedir. Daha önce de belirtildiği gibi ses içine veri saklama yöntemleri insanın işitme sisteminin esnekliklerini istismar etmektedir.

İşitme sistemi görme sistemine oranla çok daha hassas olmasına rağmen yine de veri saklama işleminde kullanılabilir esnekliğe sahiptir. 16 bitlik ses örneklerinden oluşan bir WAV dosyasına veri saklama işleminde başlık kısmının boyutu ihmal edildiğinde Low Bit Encoding yöntemiyle taşıyıcı dosyanın 1/16 sına veri saklayabilmektedir [12].

6.2. Analiz İşlemi Ve Detayları

Burada hedefim Westfeld ve Pfitzmann'ın Ki-kare saldırısıyla [58] deney yapmak ve ideal olarak sonuçlarını doğrulamaktır. Asıl amacım, burada söz edilen algoritmalardan en azından birini kullanılarak gömülmüş olan veriler içeren sesler üzerinde test etmektir. Ne var ki, Westfeld ve Pfitzmann Ki-kare saldırısının diğer gömme algoritmalarına başarılı bir şekilde saldırmak üzere değişikliğe uğratılabilir. Dolayısıyla üzerinde Ki-kare saldırısını test edeceğim basit bir algoritma oluşturdum.

Bu algoritmaları oluşturmada, bir kaç karar aşağıdaki şekilde alınmıştır:

1. Gömme algoritması piksel başına en fazla bir bit gömecektir. Bazı pikseller gömülü bitler içermeyebilir. Bu karar hem basitlik açısından hem de veriyi ses içerisine algılanamaz bir şekilde gömmek için alınmıştır.
2. Gömme algoritması bitleri, örtü ses ilk sıradan başlayarak sonundaki piksellere ve örtü sese her bir mesaj biti gömecektir. Bu gömme planının amacı büyük oranda süreci hata ayıklama için basitleştirmektir ancak aynı zamanda bir sesin çeşitli kısımlarını yani veri gömülü kısımlar ve veri gömülmemiş kısımları test etmemize izin vermektedir.
3. Her bir mesaj gömme işleminden önce rastgele olarak oluşturulacaktır. Bu tasarımın birinci amacı sonuçlarımı gömülmüş mesajın özel içeriğine atfetme olasılığını en aza indirgemektir. Eğer gömme işlemi her meydana geldiğinde mesaj değişirse ve her yeni mesaj gömüldüğünde sonuçlar uyumluysa sonuçlar doğrudur.

Bu algoritmaları oluşturmada, bir kaç varsayım aşağıdaki şekilde alınmıştır:

1. Mesajdaki bitlerin sayısı örtü sesteki piksellerin sayısına eşit ya da daha düşüktür. Bu durum, her bir pikselin en fazla bir adet gömülü biti olduğu kararını karşılamak üzere gerekli bir koşuldur.

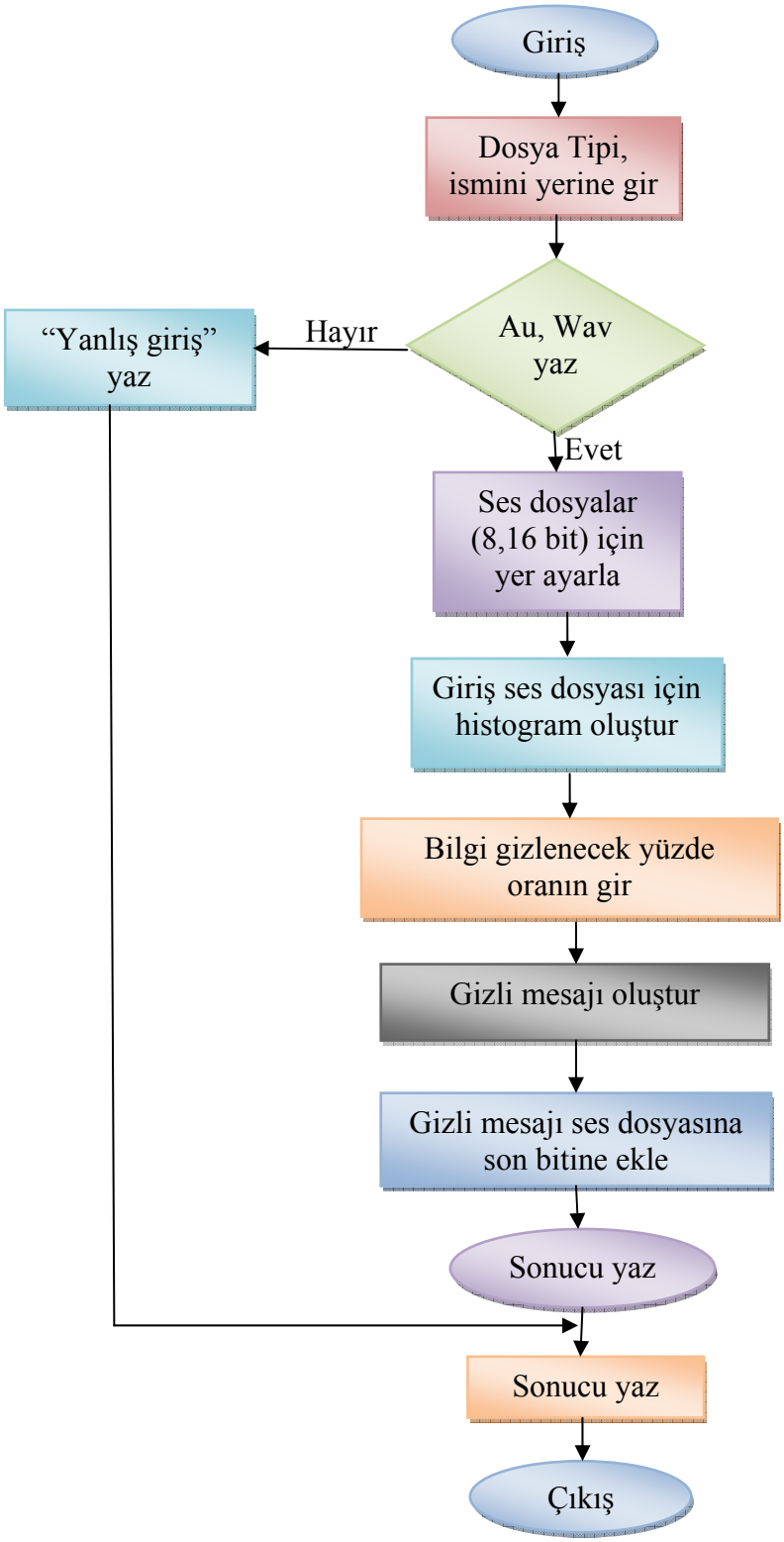
2. Mesaj 0 ve 1'lerin rastgele dağılımıdır. Genel olarak steganografi amacıyla seslere gömülen veriler gömülmeden önce şifrelenerek, böylece rastgele bir 0'lar ve 1'ler dağılımı oluşturur. Dolayısıyla gömülmekte olan verilerin şifrelenmiş olduğunu varsayıyoruz.

6.3. Ki-kare Saldırısı

Makalelerinde, Westfeld ve Pfitzmann [58] geleneksel inanışın aksine, bir görüntüdeki en az önemli bitlerin tam olarak rastgele olmadığını iddia etmektedir. Dahası, her bir değer çiftindeki iki pikselin her birinin sıklığının değer çiftinin ortalamasından uzağa düşme eğilimde olduğunu iddia etmektedirler. Diğer bir deyişle, içine veri gömülmüş tipik bir görüntü içerisinde 2k piksel değerinin sıklığının 2k + 1 piksel değerinin sıklığına (neredeyse) yakın olması yaygın olmayan bir durumdur. Dahası, bilginin değer çiftleri yaratan FE aracı gibi bir algoritma kullanılarak bir görüntüye gömülmesi nedeniyle 2k ve 2k + 1 eşit ya da neredeyse eşit hale gelir. Ki-kare saldırısı bu neredeyse eşit değer çiftlerini algılamak üzere tasarımlanmıştır ve gömme olasılığını test görüntüsündeki çift sayılı piksel değerleri ve bunların karşılık gelen tek sayılı piksel değerlerinin eşit ne kadar yakın olduğuna dayandırmaktadır.

Ki-kare saldırısı çeşitli gömme algoritmalarına uyarlanabilir olarak tasarımlanmıştır ancak temel kavram gömme algoritmasına bakılmaksızın aynıdır. Bölüm 6.4. FE aracıSes'e uygulandığı şekilde Ki-kare saldırısını tanımlamaktadır.

6.4. Veri Saklama Yaklaşımları (FE aracıSes)



Şekil 6.4. Ses tabanlı bilgi gizleme akış diyagramı

Yüksek düzey bir açıklama olarak, yazdığım FE aracıSes adlı algoritma mesaj bitine uyacak şekilde örtü sesteki her bir pikselin LSB'sine gömecektir. $c_{ij} \in \{0, 1, 2, \ldots, 255\}$ elemanlarıyla $C^{p\times q}$ 8 veya 16 bitlik gri ölçekli örtü sesini temsil etsin; $m_{ij} \in \{0, 1\}$ elemanlarıyla $M^{p\times q}$ gizli mesajı temsil etsin; ve $s_{ij} \in \{0, 1, 2, \ldots, 255\}$ elemanlarıyla $S^{p\times q}$ (8 veya 16 bitlik gri ölçekli bir ses olan) stego sesi temsil etsin. C örtü sesin kullanıcı tarafından seçilmiş ve FE aracıSes'e okutulmuştur. Varsayılan olarak, Matlab görüntüleri matrisler olarak okur ve 8 veya 16 bitlik matrisler için her bir elemanın 0 ila 255 arasındaki tam sayılardır. Sestekini okuduktan sonra, M rastgele dağıtılmış 0'lar ve 1'lerden oluşmuş bir matris olacak şekilde ilk olarak 0 ila 1 arasında değişen rastgele oluşturulmuş p × q'luk matris olarak M' matrisini oluşturur ve daha sonra M= mod([10M'], 2)'yi hesaplayarak FE aracıSes mesajı oluşturur. Kullanıcı ayrıca örtü sesteki büyüklüğünün % 1'i ila % 100 arasında değişebilecek bir mesaj uzunluğu da seçebilmektedir. Bu özelliğin amacı hem değişmiş hem de değişmemiş piksel değerleri içeren sesler üzerinde, Ki-kare saldırısını test etmemizi sağlamaktır. Her bir piksele veri gömmenin steganografi sistemi kapasitesi için makul beklentileri aşması iddia edilebileceğinden dolayı piksellerin % 100'ünden daha azına bilgi gömülü olan görüntüleri test etmek faydalı olacaktır. Dahası bir Sese gömülü durumda olan mesaj pratik durumlarda nadiren örtü sesin % 100'ünü tam olarak dolduracaktır.

```
audio_name = input('Enter name  of Main AUDIO = ','s');
audio_type = input('Enter audio type,e.g.,wav,... =
','s');
```

Yukarıdaki kod parçası kullanıcı tarafından, sesin konumunun ve adının belirlendiği yerdir. Bu bölümde geliştirilen yazılımda 2 tür ses dosyası kullanılmıştır. Bu ses dosyası türleri WAV (Wavelate) ve AU'dir (Audio Unit) . Ses dosyasının adı ve konumu belirtildikten sonra kullanıcı dosya uzantısını belirler. Yanlış giriş durumunda program hata verecektir. Yanlış girişleri tespit edebilmek için aşağıdaki gibi bir "switch" yapısı kullanılmıştır. Yanlış giriş neticesinde "Wrong input!" şeklinde bir uyarı kullanıcıya gösterilir.

```
switch audio_type

    case 'au'
        Audio = [audio_name '.' audio_type];
        [input_audio,Fs,bits] = auread(Audio);

    case 'wav'
        Audio = [audio_name '.' audio_type];
        [input_audio, Fs, bits] = wavread(Audio);

    otherwise
        disp ('Wrong input! ');
        disp (' ');
end
```

Giriş olarak belirtilen ses dosyası yazılım tarafından işlendikten sonra çıkış dosyasının nereye kaydedileceği kullanıcıya sorulur. Buradan elde edilecek sonuç dosyası daha sonra analiz yazılımında kullanılacaktır. Takip eden iki satırda yer alan kod parçası bu işlemi gerçekleştirmektedir.

```
output_audio = input('Enter name of OUTPUT
AUDIO(exclude extension):','s');

a = false;
while a == false

    N2 = input('Enter the output audio bits (8 bits or
16 bits): ','s');
    N = str2num(N2);

    switch N
        case 8
            a = true;

        case 16
            a = true;

        otherwise
            disp ('Wrong input! Please select either 8
bits(8) or 16bits(16)');
            disp (' ');
    end
end
```

Yukarıdaki kod parçasında çıkış ses dosyasının kaç bitlik olacağı kullanıcıya sorulmaktadır. Çıkış ses dosyası için 8 bit ve 16 bit olmak üzere iki seçenek mevcuttur. Kullanıcı bu iki değerden birini girdiği zaman program çalışmaya devam edecek, aksi taktirde kullanıcıya "Wrong input! Please select either 8 bits(8) or 16bits(16)" şeklinde bir uyarı verilecek ve doğru giriş yapılana kadar çıkış ses dosyasının kaç bitlik olacağı sorulmaya devam edecektir.

```
yleft=input_audio(:,1);
```

Yukarıdaki kod parçası stereo ses dosyalarını mono ses dosyaları haline çevirir. Stereo ses dosyaları yerine mono ses dosyalarını kullanmak kolaylık sağlamaktadır. Bilginin iki kanalda bulunması yerine tek kanalda bulunması analiz sonuçlarının daha iyi değerlendirilebilmesini sağlayacaktır.

```
answer=input('Histograms? y or n:','s');
```

Yukarıdaki satırda kullanıcıya giriş ve çıkış dosyalarının histogramlarını görmek isteyip istemediği sorulmaktadır. Giriş ve çıkış dosyalarının farklı olduklarının gözle görülebilmesi için histogramlar uygun grafiklerdir.

```
cover=2^(bits-1).*(yleft);
cover = round (cover);
cover = cover + (2^(bits-1));


[p,q]=size(cover);
Xa=zeros(2^(bits-1),1);
Xb=zeros(2^(bits-1),1);
Ya=zeros(2^(bits-1),1);
Yb=zeros(2^(bits-1),1);
for i=1:p
    for j=1:q
        if rem(cover(i,j),2)==0

Xb((cover(i,j)/2)+1)=Xb((cover(i,j)/2)+1)+1;
        else
            Yb(((cover(i,j)-1)/2)+1)=Yb(((cover(i,j)-
```

```
1)/2)+1)+1;
        end
    end
end

if answer=='y'
    categories= zeros(2^(bits-1),1);
    for i=1:(2^(bits-1))
        categories(i)=2*(i-1);
    end
    V=abs(Yb-Xb);
    fig_title=[audio_name,'.',audio_type,':','|Y-X|'];
    figure(1),plot(categories,V,'*:');title(fig_title);
    xlabel('category(i)'); ylabel('|X-Y|');
end
```

Yukarıdaki kod parçası giriş ses dosyası için histogramın oluşturulduğu kısımdır. Kullanıcı histogram oluşturulması için onay vermişse, oluşturulan histogram ekranda görüntülenecektir. Aksi takdirde program sonraki kısımdan çalışmaya devam edecektir. "`Figure(1)`" giriş ses dosyasının histogramını göstermektedir.

```
length=input('Enter length of desired message(in
percent--use integers 1-100 only):');
while ((length<1)|(100<length))
    if(100<length)
        length=input('Message too large for cover
image.Enter length of desired message(in percent--use
integers 1-100 only):');
    else
        length=input('Invalid message length.Enter
length of desired message(in percent--use integers 1-
100 only):');
    end
end
```

Yukarıdaki kod parçasında kullanıcıdan 1 ile 100 arasında bir sayı girmesi istenmektedir. Bu sayı, giriş ses dosyasının yüzde kaçı üzerine bilgi gizleneceğini belirtmektedir. Bu sayının istenmesi, analiz kısmında kolaylık sağlamaktadır. Giriş ses dosyasının ne kadarında bilgi saklı olduğunun bilinmesi, analiz sonuçlarının doğruluğunun test edilmesi ve analizin doğruluk oranının saptanması açısından

önemlidir. Kullanıcının 1 ile 100 arasında olmayan bir sayı girmesi durumunda kullanıcı uyarılmakta ve yeniden giriş yapılması istenmektedir.

```
length=floor((length/100)*p*q);
save('length','length');

M1=rand(p,q);
M1=rem(floor(10*M1),2);
rows=floor(length/q);
col=length-(rows*q);
stego=floor(cover/2)*2+M1;
if rows<p

nextrow=cat(2,stego(rows+1,1:col),cover(rows+1,col+1:q))
;
end
switch rows
    case p
        stego=floor(cover/2)*2+M1;
    case (p-1)
        stego=cat(1,stego(1:rows,1:q),nextrow);
    otherwise

stego=cat(1,stego(1:rows,1:q),nextrow,cover(rows+2:p,1:q
));
end
save('M1','M1');
```

Yukarıdaki kod parçası, giriş ses dosyasında saklanacak olan bilgiyi rastgele ve ikili olarak (0 ve 1) oluşturmaktadır. Sonuçta M1 isimli ve rastgele bitlerden meydana gelmiş bir tek boyutlu dizi elde edilmektedir. Burada bir hususu açıklamak faydalı olacaktır. Literatürde, gizlenecek olan bilgi gizlenmeden önce RSA, DES, 3DES vb bir şifreleme algoritması ile şifrelenmekte ve daha sonra bir resim veya ses dosyasında içerisine gizlenmektedir. Şifrelenmiş olan bilginin ikili hale çevrilmesi neticesinde oluşan bit dizisi ses ya da resim dosyasına kaydedildiği için, taşıyıcı dosyadan elde edilebilecek muhtelif bit dizilerinin ilk bakışta anlamsızmış gibi görünmesi sağlanmaktadır.

```
for i=1:p
    for j=1:q
        if rem(stego(i,j),2)==0

Xa((stego(i,j)/2)+1)=Xa((stego(i,j)/2)+1)+1;
        else
            Ya(((stego(i,j)-1)/2)+1)=Ya(((stego(i,j)-
1)/2)+1)+1;
        end
    end
end
if answer=='y'
    W=abs(Ya-Xa);
    fig_title=[output_audio,'.',audio_type,':','|Y-
X|After Embedding'];
    figure(2),plot(categories,W,'*:');title(fig_title);
    xlabel('category(i)'); ylabel('|X-Y|');
end
save('stego','stego');
```

Yukarıdaki kod parçası, rastgele elde edilmiş olan bit dizisini ses dosyasına gömmektedir. Ses dosyasının 8 bitlik veya 16 bitlik olmasına göre bilgi bitlerinin gömüleceği bitler de değişmektedir. Daha önceden kullanıcıya sorulmuş olan histogram görmek isteyip istemediğine dair sorunun cevabı "y" ise, bilgi gömme neticesinde elde edilmiş olan ses dosyasının histogramı "`Figure(2)`" olarak kullanıcıya gösterilmektedir.

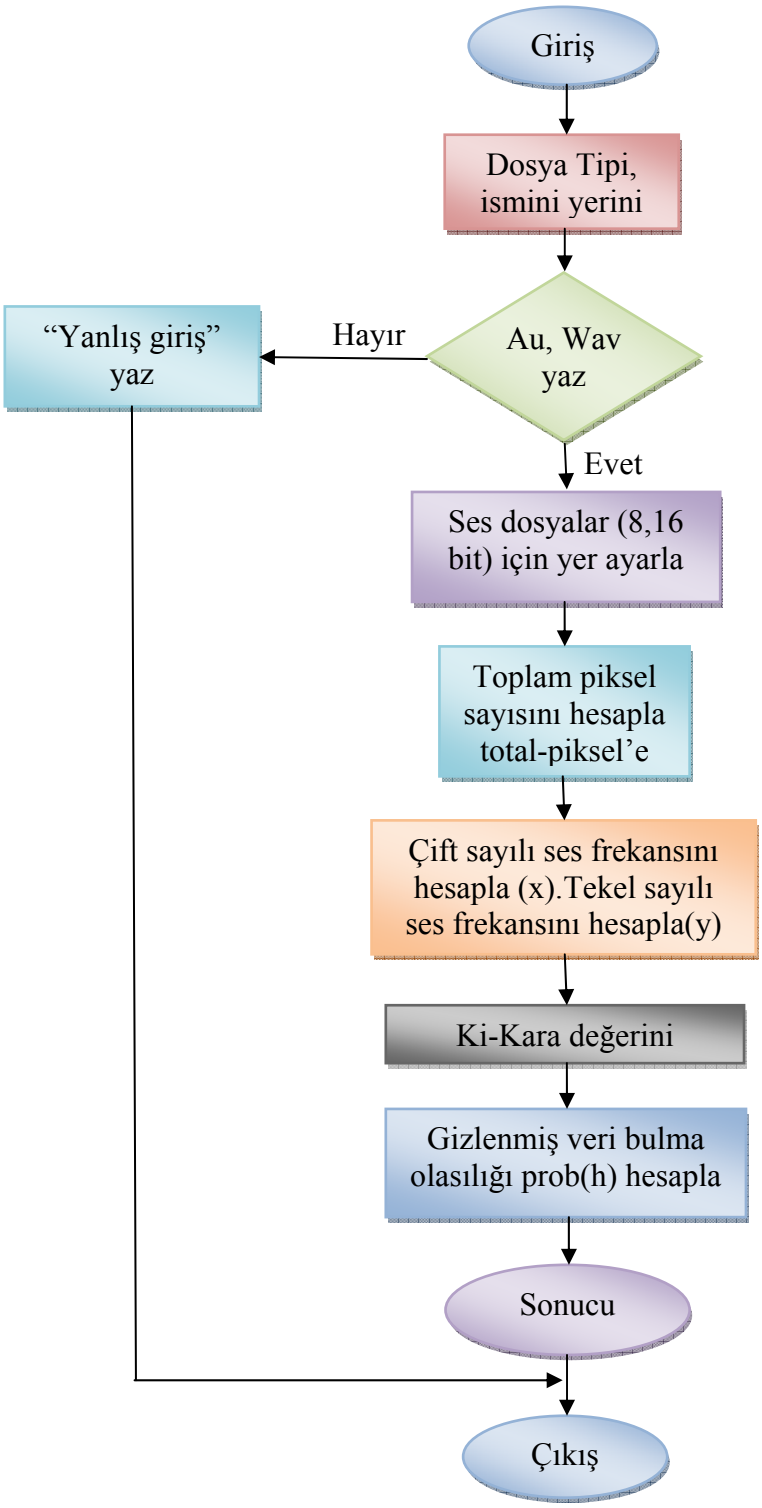
```
stego = stego - (2^(bits-1));
stego = stego ./ (2^(bits-1));
```

Yukarıdaki kod parçası, sonuçta elde edilen dosyanın 8 bitlik mi 16 bitlik mi olduğuna göre türünü değiştirmekte ve Böylece dosya doğru bir biçimde kaydedilebilmektedir.

```
switch audio_type

    case 'au'
        auwrite(stego,Fs,N,'linear',output_audio);

    case 'wav'
        wavwrite(stego,Fs,N,output_audio);

    otherwise
        disp ('Wrong input! ');
        disp (' ');
end
```

Yukarıdaki kod parçası ile sonuçta elde edilen ses dosyası, daha önce kullanıcı tarafından belirtilen konuma kaydedilmektedir. Dosyanın tipine bağlı olarak yazma fonksiyonu değişmekte olduğu için bir “switch – case” yapısı oluşturulmuştur.

6.5. Saldırıda kullanılan DÇ3Ses Algoritması



Şekil 6.5. Ses tabanlı Ki-kare saldırısı akış diyagramı

DÇses3 algılama algoritması, bir ses üzerine Ki-kare saldırısı gerçekleştirip veri gömme olasılığını çıktı olarak vermek üzere tasarımlanmıştır.

```
audio_name = input('Enter audio name and
location(exclude extension): ', 's');
audio_type = input('Enter audio type, e.g., au, wav:',
's');
switch audio_type
         case 'au'
        Audio = [audio_name '.' audio_type];
        [input_audio,Fs,bits] = auread(Audio);

    case 'wav'
        Audio = [audio_name '.' audio_type];
        [input_audio, Fs, bits] = wavread(Audio);

    otherwise
        disp ('Wrong input! ');
        disp (' ');
end
```

Yukarıdaki kod parçası daha önceden ses dosyasına gizlenmiş bilgi konumunu da belirlediğimiz kullanıcı tarafından seçilmektedir, sesin konumunun ve adının belirlendiği yerdir. Bu ses dosyası türleri WAV (Wavelate) ve AU' dir (Audio Unit) . Ses dosyasının adı ve konumu belirtildikten sonra kullanıcı dosya uzantısını belirler. Yanlış giriş durumunda program hata verecektir. Yanlış girişleri tespit edebilmek için "switch" yapısı kullanılmıştır. Yanlış giriş neticesinde "Wrong input!" şeklinde bir uyarı kullanıcıya gösterilir.

```
A=2^(bits-1).*(input_audio);
A = round (A);
A = A + (2^(bits-1));
```

Uzantı doğrulama işi gerçekleştirildikten sonra girilen ses dosyasının 8 veya 16 bitlik olmalarına göre "A" değişkeninde yer ayırtılıyor.

```
[m,n] = size(A);
```

Her türlü boyuttaki ses dosyalarını tüm piksellerini ele almak için “m” ve “n” değişkenleri “FOR” döngüsünden geçirilmesini sağlamaktadır.

```
percentage = zeros(101,1);
prob = zeros(101,1);
K = zeros(101,1);
```

Yüzdelerin Bir ile Yüz değerlerini aşmamak için yukarıdaki üç aşama gerçekleşir.

```
for h=1:100
    k = 2^(bits-1);
    percentage(h) = percentage(h) + h;
    total_pixels = floor((h/100)*m*n);
```

Hesaplaşmalarda gerçek sayının kullanımını garanti edebilmek için yüzdeleri gerçek sayıya çevirириz sonra tüm pikselleri hesaplayarak test ederiz.

```
whole_rows = floor(total_pixels/n);
    columns_last_row = total_pixels - (whole_rows*n);
```

FOR döngüsünde kullanmak üzere gereken total_pixel sayısını hesaplamak için satır sayısını hesaplarız. Sonra gerektiği (hesapladığımız satırlardaki) sülünlerdeki pikselleri hesaplarız.

```
X=zeros(2^(bits-1),1);
Y=zeros(2^(bits-1),1);
D=zeros(2^(bits-1),1);
```

2^7 ya da 2^{15} boyutundaki matrisleri sıfırlarız. Burada X çift sayılı ses frekansını hesaplar, Y tekel sayılı ses frekansını hesaplar ve D Ki-kare varsaymaları hesaplarında yardımcı olur.

```
for q=1:whole_rows
        for r=1:n
            if rem(A(q,r),2)==0
                X((A(q,r)/2)+1)=X((A(q,r)/2)+1)+1;
            else
                Y(((A(q,r)-1)/2)+1)=Y(((A(q,r)-
1)/2)+1)+1;
            end
        end
    end

    for q=whole_rows+1
        for r=1:columns_last_row
            if rem(A(q,r),2)==0
                X((A(q,r)/2)+1)=X((A(q,r)/2)+1)+1;
            else
                Y(((A(q,r)-1)/2)+1)=Y(((A(q,r)-
1)/2)+1)+1;
            end
        end
    end

    Z = (X + Y);

    for i=1:(2^(bits-1))
        if (X(i)+Y(i)) < 5
            X(i) = 0;
            Y(i) = 0;
            k = k-1;
        end
    end
```

Girdi ses dosyanın piksellerinin frekans sayılarını Döngüde hesaplayarak: Piksel değeri çift ise (yani Piksel değeri = 2i+1) o zaman X'teki (i+1) sayılı satır bir artar. Piksel değeri tekel ise (yani Piksel değeri = 2i) o zaman Y'deki (i+1) sayılı satır bir artar.

```
Z=Z/2;
```

X_i ve Y_i değerlerinin ortalamasını kaydeden matris (Z_i = X_i + Y_i) /2)

```
D = (X-Z).^2;
for i=1:(2^(bits-1))
        if Z(i)==0
            D(i) = 0;
        else
            D(i) = D(i)./Z(i);
        end
    end

    format long g;
    C=sum(D);
```

C burada Ki-kare istatistiğine işaret etmektedir.

```
Pcheck=1-gammainc(C/2,(k-1)/2);
```

gammainc fonksiyonun kullanarak olasılığı hesaplarız.

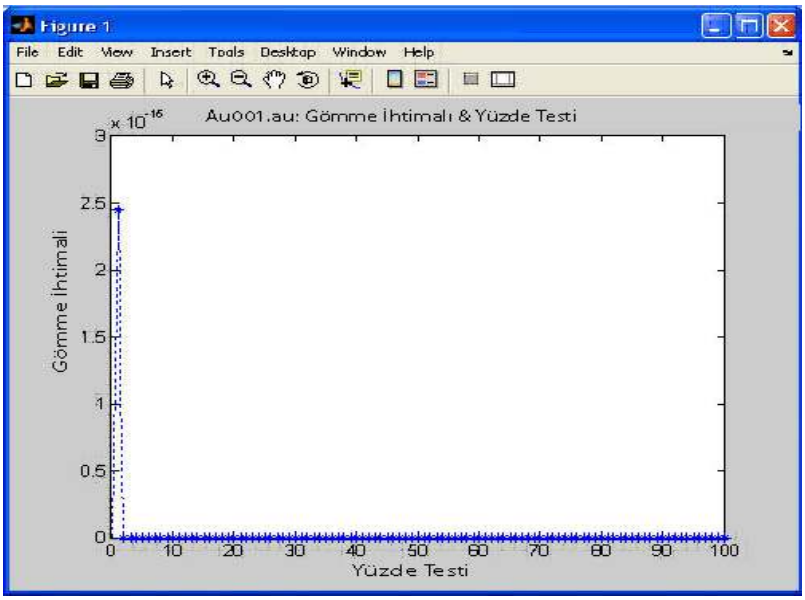
```
    prob(h) = prob(h) + Pcheck;
    K(h+1) = K(h+1) + k;
end

fig_title = [audio_name, '.',
audio_type,':','Probability of Embedding vs.Percentage
Tested'];
percentage = cat(1, percentage(101),percentage(1:100));
prob = cat(1, prob(101), prob(1:100));
result = cat(2, percentage, prob, K);
display(result);
save('result','result');
figure (1), plot(percentage,prob,'*:');
title(fig_title); xlabel('Percentage Tested');
ylabel('Probability of Embedding');
```

ve en son sonuçları kayıt etmektedir.

6.7. Deneysel Sonuçlar

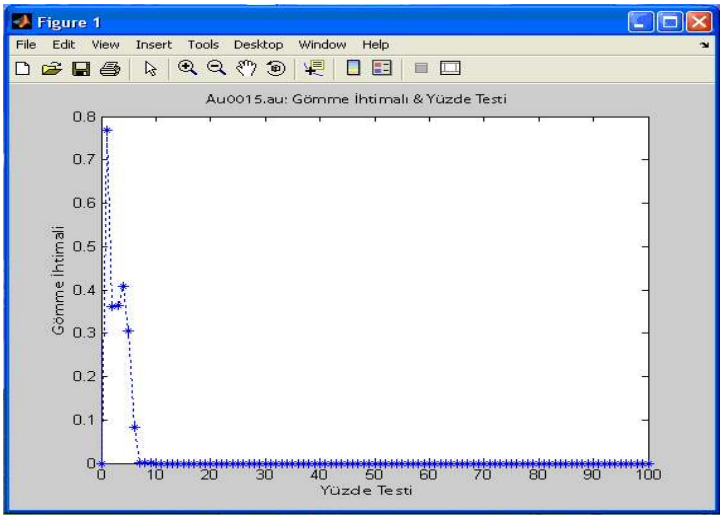
Test etmek üzere beş adet Wavelet(WAV) dosyası ve beş adet Audio Unit (AU) değişik büyüklük ve düzgünlük ölçüsünde ses seçtim. Orijinal sesler "Wav001", "Wav002"...."Wav005" ve "Au001", "Au002"...."Au005" olarak etiketlenmiştir. Her birine, piksellerin %10'u,%50'si ve %100 oranında bilgi gömdüm ve bu stego sesleri her birini örneğin Wav002 içerisindeki piksellerin %10'una bilgi gömülmesi "FlipWav00210" ve Au002 içerisindeki piksellerin %10'una bilgi gömülmesi "FlipAu00210" adlı stego ses oluşturacak şekilde " Flip[ses adı][bilgi gömme yüzdesi]" formatında kaydettim. 15 stego WAV ve 15 AU oluşturduktan sonra bu stego sesleri her birini ve orijinal ses (toplam 20 WAV ve 20 AU ses) DÇses3'ü kullanarak test edilmiştir.

Seçtiğim seslerin örneklemesi küçük (yalnızca 5WAV 8bit ses dosyası ve 5AU 8bit ses dosyası) olmasına rağmen, bazı ilginç sonuçlar elde ettim. İlk olarak DÇ3ses, test ettiğim seslerin yalnızca yarısından fazla özellikle büyük homojen bölgeleri onları doğru bir şekilde analiz ediyor gibi görünmektedir. Örneğin Au001 bu seste,DÇ3ses neredeyse sıfıra yakın gömme olasılığı verdiği piksellerin ilk %2'si haricinde DÇ3ses doğru bir şekilde algılamaktadır Şekil 6.6.



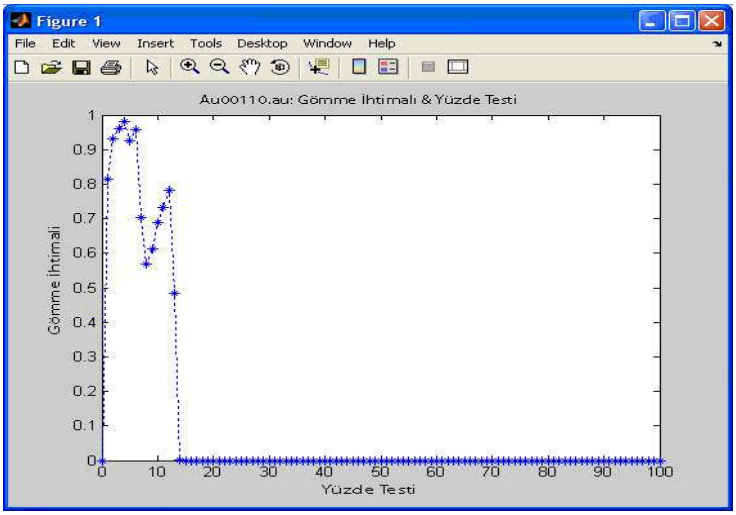
Şekil 6.6. (Au001)Orijinal Sese Ait Ses Birim Analizi (%0 veri gizlenmiş)

Şekil 6.7'de piksellerin %5'sine bilgi gömülmüş olan Au001 olan FlipAu0015'nin verilerini göstermektedir. Gömülen olasılığının piksellerin %7'i test ittikten sonra sert bir şekilde düştüğüne dikkat ediniz. Benzer düşmeler diğer görüntülerde de meydana gelmiştir.

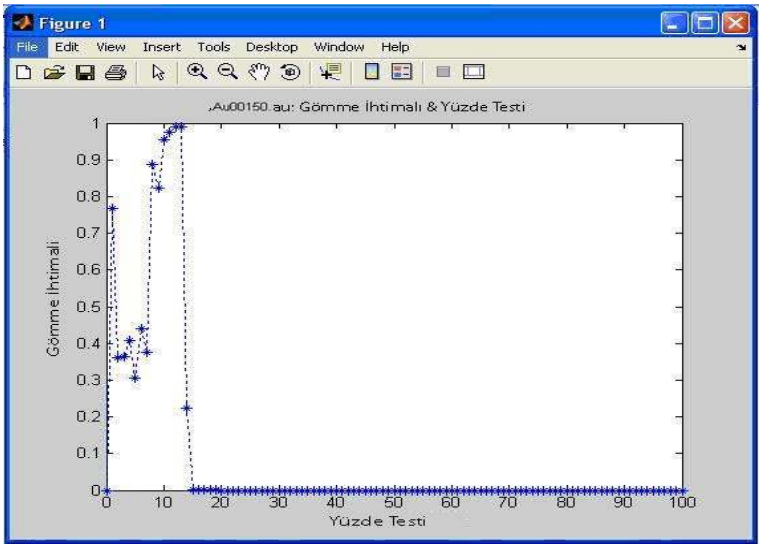


Şekil 6.7. (Au0015)Stego Sese Ait Ses Birim Analizi (%5 veri gizlenmiş)

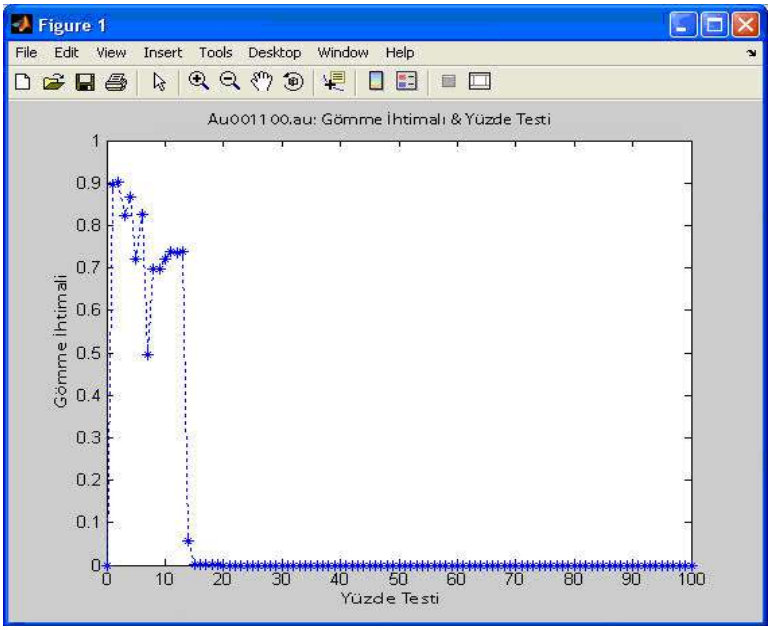
FlipAu00110'da, Pr, r > 13 olduğunda düşmekte ve FlipAu00150 ve FlipAu001100'de Pr, r > 15 olduğunda düşmektedir. (Bakınız Şekil 6.8, Şekil 6.9 , Şekil 8.10). Burada saldırının yanlış sonuncu, açık bir şekilde, yeterli rastgelelik gösteren örtü ses seçilmesi ki-kare testini özünde yararsız hale getirir.



Şekil 6.8. (Au00110)Stego Sese Ait Ses Birim Analizi (%10 veri gizlenmiş)



Şekil 6.9. (Au00150)Stego Sese Ait Ses Birim Analizi (%50 veri gizlenmiş)

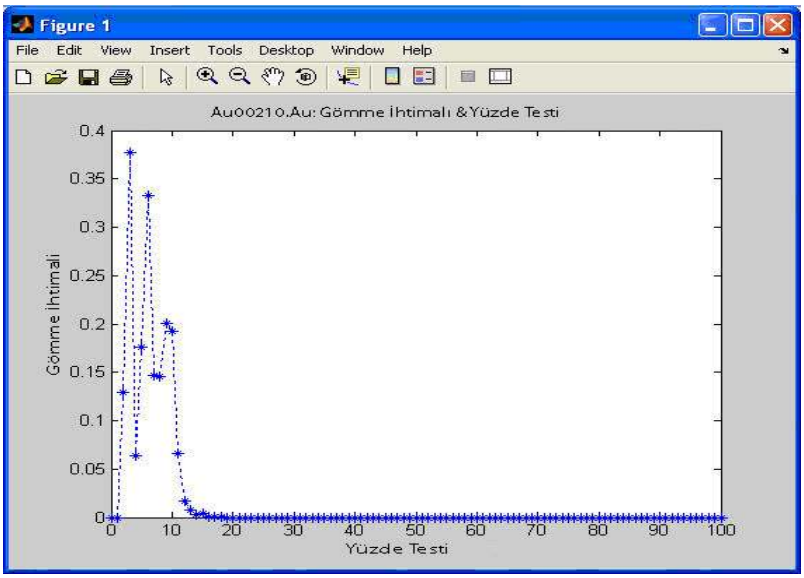


Şekil 6.10. (Au001100)Stego Sese Ait Ses Birim Analizi (%100 veri gizlenmiş)

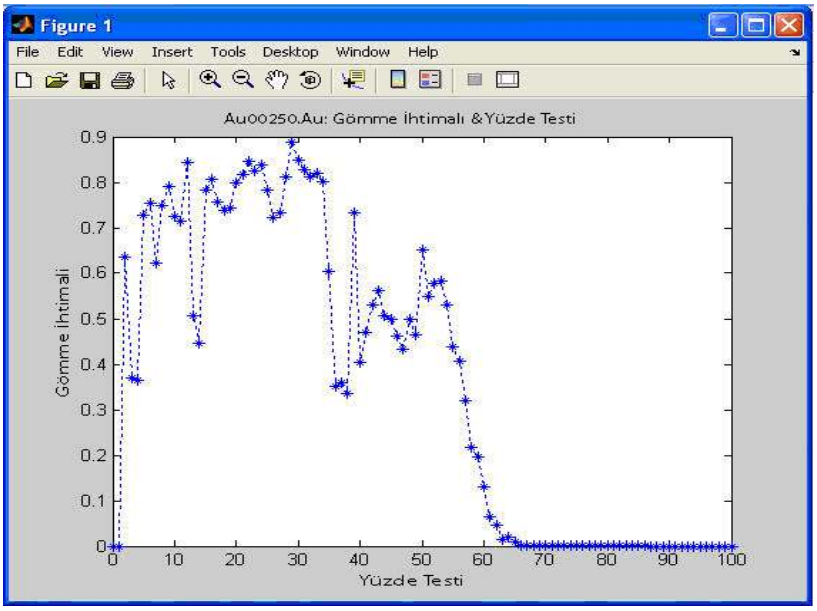
Bununla birlikte Au002, Au005 ve Wav001, Wav002'deki veri gömme sonuçları daha az güvenilir.

Au002deki piksellerden %10'u gömülü bitler içerdiğinde, düşme r = 13 iken meydana gelir ancak piksellerin %50'sine bitler gömüldüğünde, düşme r = 65'e kader meydana gelmemektedir Şekil 6.11 ve Şekil 6.12.

Dolayısıyla %50'den %65'e kadar olan pikseller sergilemektedir. Bu sonuçlar uymakta ancak veri gömülmemiş bir sesin bölgelerinde teorik olarak beklenenin aksidir.

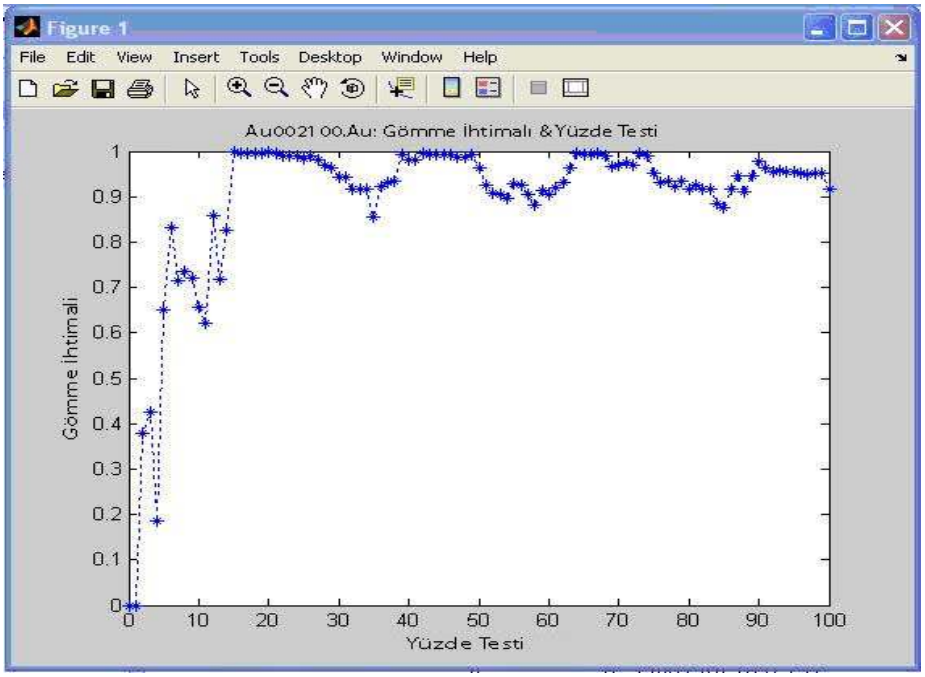


Şekil 6.11. (Au00210)Stego Sese Ait Ses Birim Analizi (%10 veri gizlenmiş)



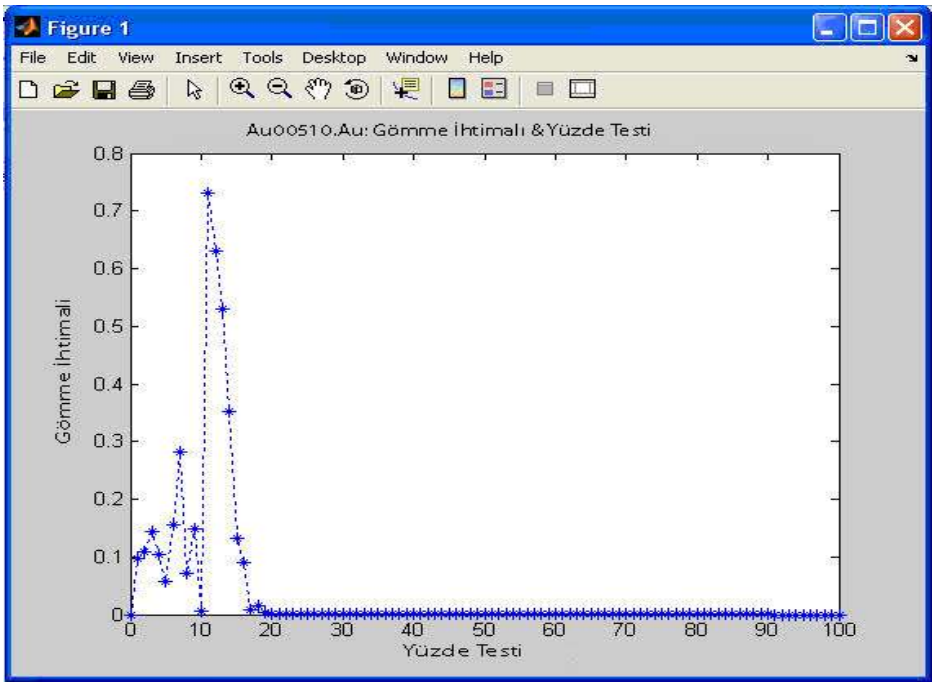
Şekil 6.12. (Au00250)Stego Sese Ait Ses Birim Analizi (%50 veri gizlenmiş)

Au002'deki piksellerden %100'una veri gömüldüğünde Şekil 6.13 yüzde 1,2 haricinde doğru bir sonuç elde etmektedir.



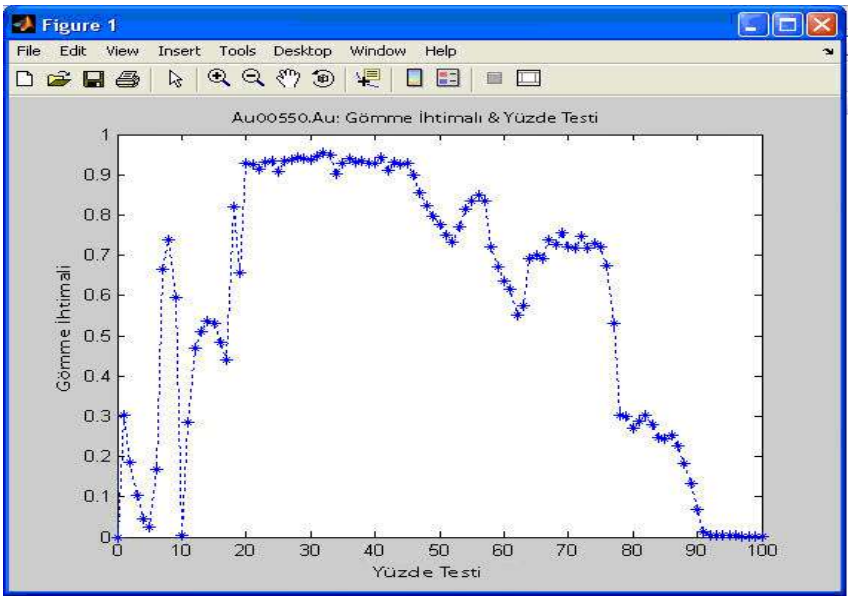
Şekil 6.13. (Au002100)Stego Sese Ait Ses Birim Analizi (%100 veri gizlenmiş)

Au005'in piksellerinin %10'una bitlerine gömüldüğünde Şekil 6.14, Pr $0 \leq r \leq 10$'ın sıfıra yakın, ondan sonra yükselmektedir yani bire yakın oluyor tekrar r = 18 sıfıra düşmektedir.



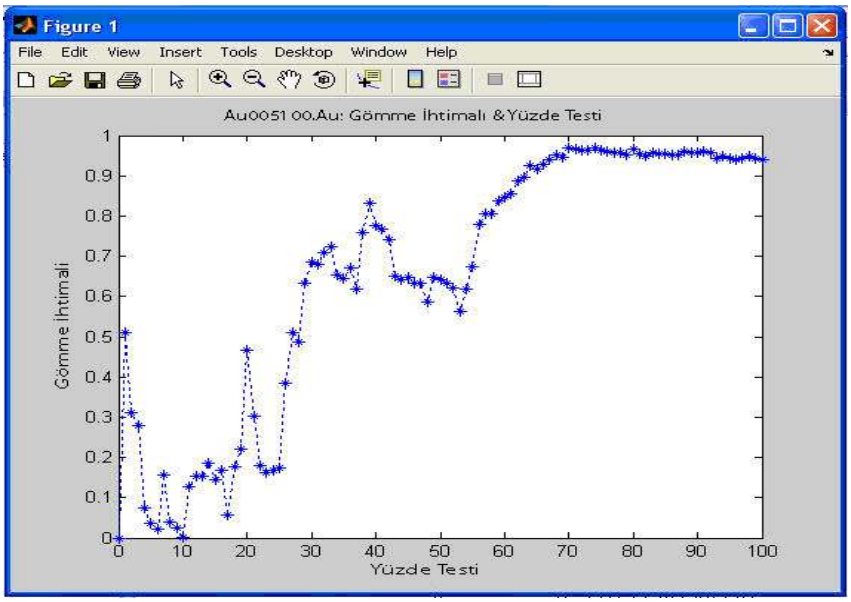
Şekil 6.14. (Au00510)Stego Sese Ait Ses Birim Analizi (%10 veri gizlenmiş)

Au005'taki piksellerin %50'sine veri gömüldüğünde Şekil 6.15, Pr yine $0 \leq r \leq 10$ için sıfıra yakın ancak $20 \leq r \leq 80$ için bire çok yakın olup, bu noktadan sonra ve $r = 90$ sıfıra düşmektedir.



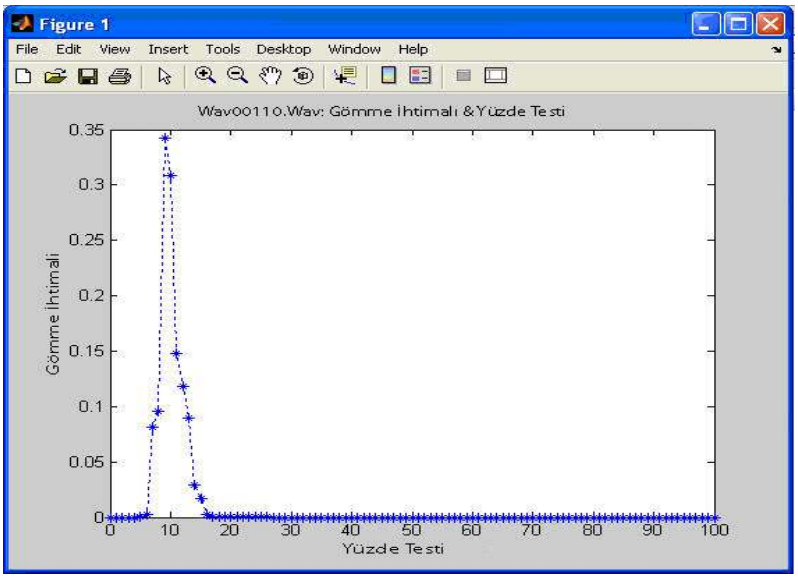
Şekil 6.15. (Au00550)Stego Sese Ait Ses Birim Analizi (%50 veri gizlenmiş)

Au005'taki piksellerin %100'una veri gömüldüğünde Şekil 6.16, Pr $0 \leq r \leq 30$ için sıfıra yakın ancak $30 \leq r \leq 100$ için bire çok yakın olmaktadır.



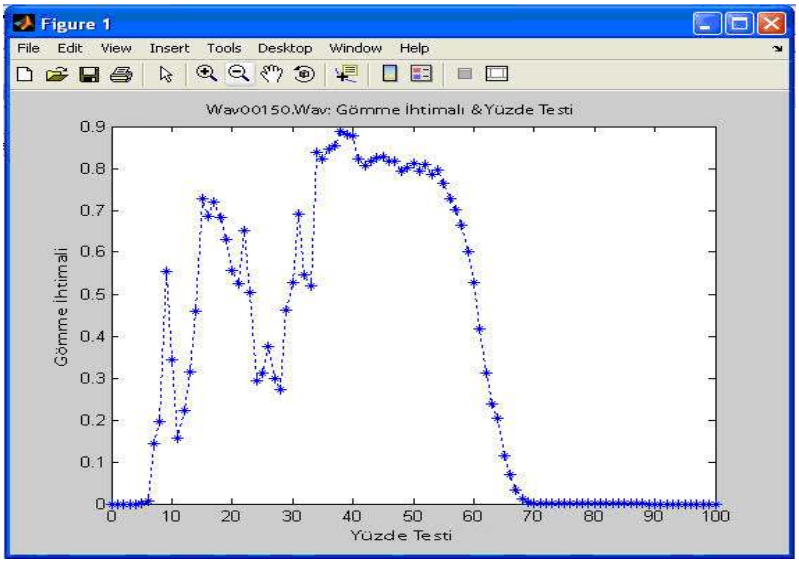
Şekil 6. 16. (Au005100)Stego Sese Ait Ses Birim Analizi (%100 veri gizlenmiş)

Wav001'deki piksellerin %10'una bitler gömüldüğünde Şekil 6.17, Pr $0 \leq r \leq 8$i.in sıfırdır ondan sonra yükselmektedir, tekrar r = 17'de sıfıra sert bir şekilde düşmektedir.



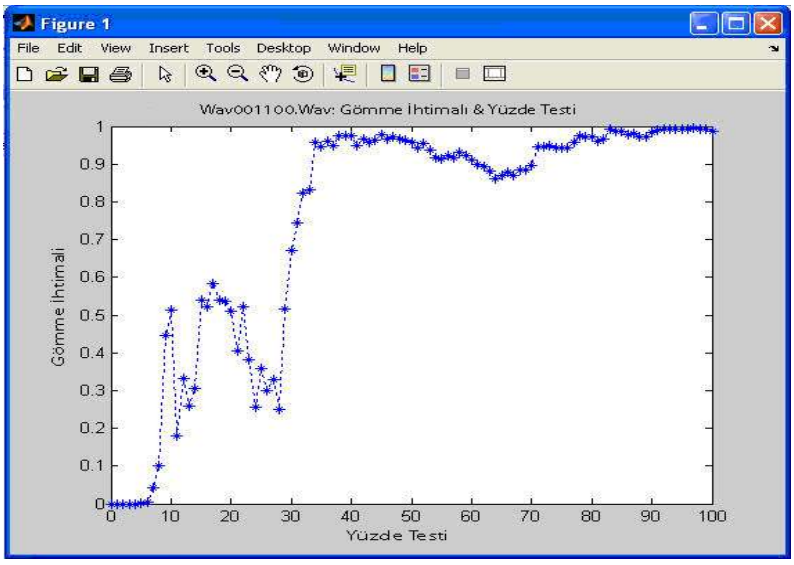
Şekil 6.17. (Wav00110)Stego Sese Ait Wav Analizi (%10 veri gizlenmiş)

Wav001'deki piksellerin %50'sine bitler gömüldüğünde Şekil 6.18, Pr yine $0 \leq r \leq 8$i.in sıfırdır ondan sonra yükselmektedir, tekrar r = 68'de sıfıra sert bir şekilde düşmektedir.



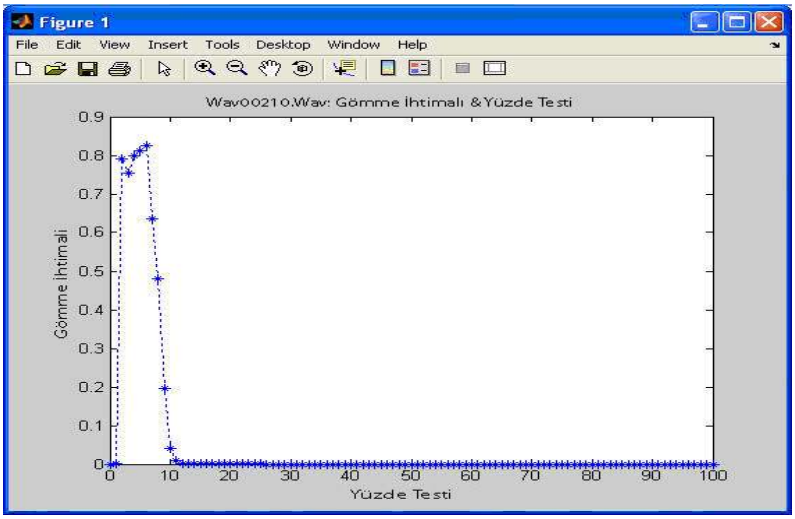
Şekil 6.18. (Wav00150)Stego Sese Ait Wav Analizi (%50 veri gizlenmiş)

Wav001'deki piksellerin %100'una bitler gömüldüğünde Şekil 6.19, Pr yine $0 \leq r \leq$ 8i.in sıfırdır ve $8 \leq r \leq 30$ sıfıra yakın ondan sonra $r \geq 30$ bire yakın olup ve öyle devam etmektedir.



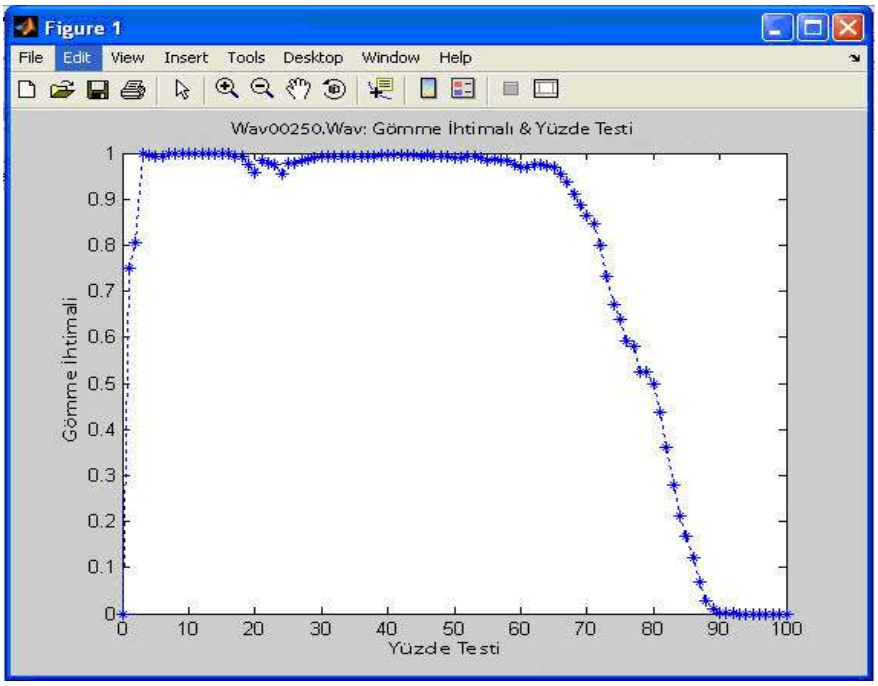
Şekil 6.19. (Wav001100)Stego Sese Ait Wav Analizi (%100 veri gizlenmiş)

Wav002'deki piksellerin %10'una ve %100'una veri gömüldüğünde Şekil 6.20 ve Şekil 6.21, doğru bir sonuç elde etmekteyiz yalnızca piksellerin %50'sine veri gömüldüğünde r = 50 olması beklenen, düşme r = 89'e kader meydana gelmemektedir Şekil 6.22.

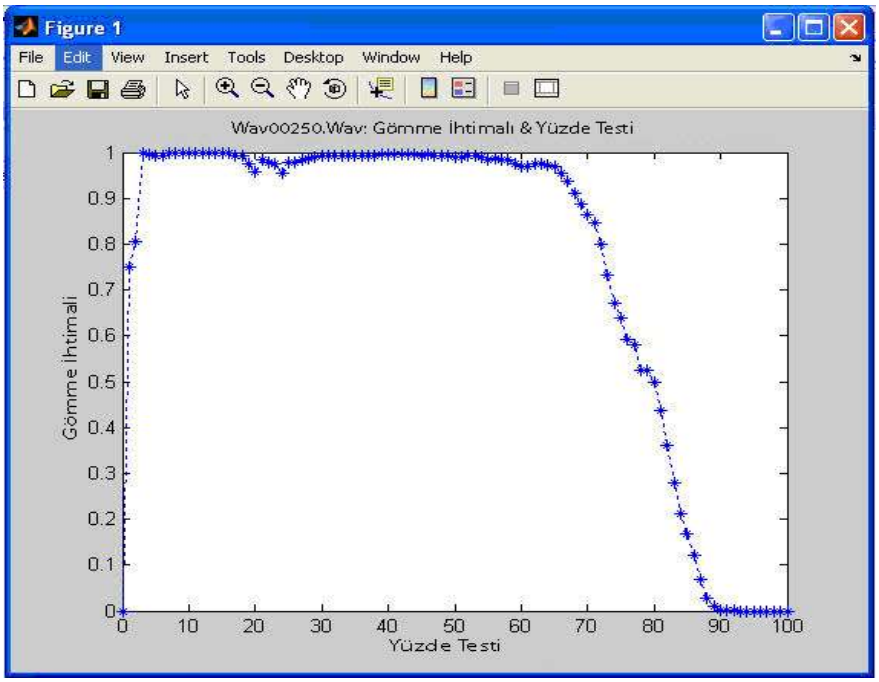


Şekil 6.20. (Wav00210)Stego Sese Ait Wav Analizi (%10 veri gizlenmiş)

Dolayısıyla %50'den %89'e kader olan pikseller bir stego sesinde beklenenlere benzer özellikler sergilemektedir. Bu sonuçlar uymakta ancak veri gömülmemiş bir sesin bölgelerinde teorik olarak beklenenin aksinedir.



Şekil 6.21. (Wav00250)Stego Sese Ait Wav Analizi (%50 veri gizlenmiş)



Şekil 6.22. (Wav002100)Stego Sese Ait Wav Analizi (%100 veri gizlenmiş)

6.8. Ki-kare Saldırısının güvenilirliği

Yanıtlamaya başladığımız sorulardan birisi "Bir görüntünün gömülü bilgi içerip içermediğini söyleyebilir miyiz"'di ve basit yanıt "Bazen"'dir. Test edilen her bir ses dosyasını piksellerinin % 100'üne veri gömüldüğünde, Ki-kare testinin bire çok yakın bir veri gömme olasılığı geri döndürmektedir. Diğer bir deyişle, (bu çalışmada) piksellerinin % 100'üne veri gömülmüş herhangi bir ses dosyasını test edildiğinde, Ki-kare testi her bir ses dosyası için gerçek pozitif sonuç geri döndürmektedir. Çizelge 6.1 ve Çizelge 6.2'de piksellerin % 100, 50%, 10%, 0%'na veri gömülmüş ses dosyaların için Ki-kare saldırısına göre düşmeler verilmiştir.

Çizelge 6.1. Ki-kare saldırısına göre "Düşmeler"(Audio Unit)

Ses dosyaları veri gömme yüzdesi	Au1	Au2	Au3	Au4	Au5
% 0	2/0	0	0	0	0
% 10	13	12	11	11	19
% 50	14	65	65	53	90
% 100	14	2/100	100	100	100

Çizelge 6.2. Ki-kare saldırısına göre "Düşmeler"

Ses dosyaları veri gömme yüzdesi	Wav1	Wav2	Wav3	Wav4	Wav5
% 0	0	0	1-2/0	2-4/0	2/0
% 10	7/18	10	11	11	2/12
% 50	7/69	89	52	55	2/53
% 100	7/100	100	100	100	2/100

Bir Ses dosyası içerisinde yeterli veri bulunduğunda, Ki-kare testi bunların varlığını algılayabilir ancak test gürültülü Ses dosyalarında yada Ses dosyalarında gürültülü olan bölgeleri için yanlış pozitif değerler geri döndürebilir. Sonraki mantıksal soru "Her hangi bir Ses dosyasına ne kadar veri gömüldüğünü söyleyebilir miyiz?"'dir. Bunu yanıtlamak için r p_r'nin düşmesine dikkat ederek gömme işleminin nerede durduğunu yüzde bir kaçlık bir hassasiyetle söyleyebilmekteyiz. Gömülmüş veriden sonraki pikseller test edildiğinde p_r sert bir şekilde düşmektedir. Bununla birlikte test etmiş olduğum çoğu ses dosyaları için bu durumun doğru olduğunu tespit etmedim.

Dolayısıyla 5WAV ve 5AU Ses dosyaları için düşmelerin nerede meydana geldiğini gösteren Çizelge 6.1 ve Çizelge 6.2, hem p_r'nin 0,90'nın altına düştüğü hem de p_r'nin altına düştüğü yerleri göstermektedir.

Daha önce dikkat çekildiği gibi, Ki-kare testi daima Ses dosyaların içerisine ne kadar veri gömülü olduğunu yüksek doğrulukla göstermemektedir. Hem Wav00150 hem de Wav00250 piksellerin % 50'si veri içermelerine rağmen piksellerinin %69 ve %89'ünde veri içermektedir. Benzer bir şekilde Au00150 ve Au00550 piksellerin %14 ve %90'nı test edilene kader düşmektedir. Ki-kare testi piksellerin %50'si veri içerdiğinde on Ses dosyasında (Wav,Au) Wav'da birinci ve ikiyinci, Au'da birinci ve beşincide yalnızca üçünde ne kadar veri gömülü olduğunu doğru bir şekilde göstermektedir.

Hata oranı = % Bulunma yüzdesi - % Veri gömme oranı

Çizelge 6.3. Wav Ses dosyasının hata oranı

Veri gömme yüzdesi	Au1	Au2	Au3	Au4	Au5
% 0	% 2	% 0	% 0	% 0	% 0
% 10	% 3	% 2	% 1	% 1	% 9
% 50	% 36	% 15	% 15	% 7	% 40
% 100	% 86	% 2	% 0	% 0	% 0

Çizelge 6.4. Au Ses dosyasının hata oranı

Veri gömme yüzdesi	Wav1	Wav2	Wav3	Wav4	Wav5
% 0	% 0	% 0	% 1	% 2	% 2
% 10	% 15	% 0	% 1	% 1	% 4
% 50	% 25	% 39	% 2	% 5	% 5
% 100	% 7	% 0	% 0	% 0	% 2

7. SONUÇ VE TARTIŞMA

Son yıllarda bilgisayar sistemlerinin güvenliği oldukça önemli bir konu olarak karşımıza çıkmaktadır. İnternet yaygınlaşmasıyla veri alış verişi ve paylaşımda artmıştır. Metin, ses, resim vb. gibi birçok veriyi içeren dosyalar, etkin bir şekilde dünyanın birçok yerindeki insanlar tarafında paylaşabilir hale gelmiştir. Bu sayede dijital ortamların içine gönderilmek istenilen bilgilerin gizlenip diğer kişilere aktarması oldukça kolaylaşmıştır. Fakat bu yöntemin kötü amaçlı kişiler tarafından kullanılması toplum ve çevre güvenliğini tehlikeye sokmaktadır. Bu nedenle dijital ortamdaki verilerin içinde gizli bilgi olup olmadığını incelenmesi oldukça önemli bir konu haline gelmiştir. Bunu sezebilmek için çeşitli steganaliz yöntemler geliştirilmiştir.

Bu tez çalışmasının gerçekleştirilmesi için ses ve resim tabanlı bir yazılım geliştirilmiş ve testler başarıyla ve kolaylıkla gerçekleştirilmiştir.

Bu tez çalışımında sunulan sonuçlarda steganografi çalışmaların daha başarılı olması için aşağıdaki değerlendirmeler yapılabilir:

1. Taşıyıcı olarak kullanılacak resim veya ses dosyasının bilinen orijinali kolay elde edilebilir dosyalar olmaması gerekir. Görsel veya işitsel steganaliz ataklarına karşı bu gerekli olabilir.

2. Renkli resimlere uygulanabilen RQP (Raw Quick Pairs) gibi bazı istatistiksel analizler, gri seviyeli resimlere uygulanmaktadır.

3. Renkli resimlerde renk çeşitliliği fazla olan sıkıştırılmamış yüksek çözünürlüğe sahip resimler daha uygun olmaktadır.

4. Saklama oranı, saklama kapasitesini düşürecek olmasına rağmen istatistiksel steganaliz karşısındaki dayanıklılığı arttırması açısından yeterince düşük tutulmalıdır.

5. Saklama işlemi ardışık değil belirli bir anahtar değerine göre rastgele yapılmalıdır.

6. Steganografi yöntemler şifreleme yöntemleri ile birlikte kullanılarak daha güvenli bir sistem oluşturulabilir.

7. Saklanacak verinin miktarı arttırılmak istenirse de gizlenecek verinin gizleme işleminden önce sıkıştırılmasıyla bu sağlanabilmektedir.

8. Steganaliz konusunda bu çalışma kapsamında anlatılanlar ve sunulan deneyler çerçevesinde Ki-kare saldırısı büyük miktarlarda veri gömüldüğünde en az önemli biti (LSB) değiştiren veri gömme algoritmalarına karşı oldukça etkilidir.

9. Ki-kare saldırısının sınırları Westfeld ve Pfitzmann'ın iddia ettiğinden daha fazla gibi görünmektedir. Örneğin Ki-kare saldırısı kendi gürültüsüne sahip bir görüntüye ne kadar veri gömüldüğünü güvenilir bir şekilde belirleyememektedir. Dahası, algoritma gürültülü bir görüntünün gömülü veri içerip içermediğini her zaman belirleyememektedir. Mutlak anlamda hiç bir veri içermeyen görüntüler için bile yanlış pozitif sonuç verebilir. Ki-kare saldırısının örneğin FE aracı gibi en az önemli biti (LSB) düz bir şekilde değiştiren algoritmalar için güvenilir bir istatistik geri döndürmedeki kısıtlamalarına ek olarak, bir çift içerisinde yalnızca her n'inci piksel değerine veri gömülmesi ya da yalnızca en az önemli biti yerine piksel değerlerinin azaltılarak değiştirilmesi gibi daha karmaşık hale gelmekte ve bu saldırının sınırları katlanarak artmaktadır.

10. Tüm dünyada olduğu gibi Türkiye'de de steganaliz konusu henüz tam olarak gereken ilgiyi görmemektedir. Genelde steganaliz özelde ise ses steganaliz uygulamalarında henüz yeteri ilerleme sağlanmamıştır. Veri transferi teknikleri hızlandıkça resim dosyalarına göre daha büyük boyutlu olan ses dosyalarının da sayısal iletişimde çok yoğun bir şekilde kullanılacağı düşünülmektedir. Bu çalışmada en yaygın ses dosyası türlerinden birisi olan PCM biçimindeki WAV ve Audio Unit (AU) dosyaları kullanılmıştır ve gerçekleştirilen yazılımın

Türkiye'de gerçekleştirilecek ses tabanlı steganaliz uygulamalara temel oluşturarak katkı sağlayacağı değerlendirilmektedir. Bu tez çalışmasında gerçekleştirilen resim tabanlı ve ses tabanlı yazılımın, çalışmadan elde edilen sonuçlar ışığında kişisel ve kurumsal güvenliğin daha farklı ve yüksek oranda sağlanmasına duyulan ihtiyacı karşılamaya yönelik olabileceği öngörülmektedir.

Bu çalışma sırasında birçok güçlükle karşılaşılmıştır. Öncelikle Türkiye'de steganaliz konusunda bir tez çalışmasının olmaması, insanlar arasında böyle bir çalışmaya ihtiyaç da bulunmadığı şeklinde yanlış bir kanı doğurduğu gözlemlenmiştir. Bu konu hakkında dolma bilgi sahibi olan kişilerin tez konusu olarak seçilebilecek rahat bir alan nitelemeleri başlangıçta olumsuz bir motivasyona sebep olmuştur. Fakat konu bütün kapsamı ile değerlendirilip, olması gereken çerçeveye oturtulunca ve çalışmanın ana hatları ortaya çıkınca, sanılanın aksine konunun oldukça kapsamlı, önemli kavram ve yeni teknolojiler ele yakından ilişkili, çok boyutlu ve devingen bir konu olduğu fark edilmiştir.

Bu tez çalışımın bundan sonra bu alanda çalışma yapacakların da yardımcı olması beklenmektedir.

Bu tez çalışmasında açıklanan yazılımların İngilizce adları yanı sıra, bunlara karşılık olarak kullanılabilecek özgün Türkçe ifadeler belirlenip kullanılmıştır.

Tez çalışması sırasında yeterli yerli ve yabancı akademik çalışmaya rastlanmaması karşılaşılan en önemli güçlüklerden biridir. Konunun dinamik bir yapı taşıması sebebi ile İnternet üzerinden çok sayıda kaynak araştırılmış ve bunların arasında en doğru ve güncel bilgi içeren kaynaklar değerlendirmeye alınmıştır. Yine yararlanılan kaynakların hemen hemen tamamının İngilizce olması, belirli bir İngilizce seviyesine sahip olunduğundan her ne kadar aşırı zorluk çıkarmasa da bunların daha doğru değerlendirilmesi için biraz daha fazla çaba harcanması gerektiği değer bir gerçektir.

11. Elde edilen sonuçlarda resim tabanlı steganaliz kullanıldığında yedi resimde iyi sonuç elde edilmiştir. Fakat gürültülü resimlerde Ki-kare saldırısı yetersiz kalmıştır. Ses tabanlı steganalizde geliştirilen yöntemde ise çok başarılı sonuçlar elde edilmiştir. Ancak, resim steganalizde olduğu gibi gürültülü seslerde öngörü içeren sesler için iyi bir sonuçlar elde edilememiştir.

KAYNAKLAR

1. Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G., "Information Hiding–A Survey", ***Proceedings of the IEEE***, Special Issue on Protection of Multimedia Content, 87(7):1062-1078 (1999).

2. Murray, A.H., Burchfiled, R.W (eds.), "The Oxford English Dictionary: Being a Corrected Re-issue", ***Oxford,*** England: Clarendon Press, (1933).

3. Phan, R.C.W., Ling, H.C., "Steganalysis of Random LSB Insertion Using Discrete Logarithms Proposed At Cita03", ***MMU International Symposium on Information and Communication Tecnologies/M2USIC 2003***, Petaling Jaya, Malaysia,, 2-3 (2003).

4. Westfeld, A., Pfitzmann, A., "Attacks on Steganographic Systems", ***Proceedings of the Third International Workshop Information Hiding, Dresden, Germany,*** 61-76 (2000).

5. Provos, N., "Defending Against Statistical Steganalysis", ***10th USENIX Securit Symposium,*** Washington, 323-335 (2001).

6. İnterent: Sellars, D., "An Introduction to Steganography", Student Papers, (1999). http://www.cs.uct.ac.za/courses/CS400W/NIS04/papers99/dsellars/index.html

7. İnternet: Lillard, T.V., "Stego forensics techniques", http://www . tlillardconsulting .com /images/DoD 2003 CyberCrime Conference Stego Forensics.ppt.(2003)

8. İnternet: Provos, N., "Steganography detection with stegdetect", http://www.outguess. org /detection .php. (2004).

9. İnternet: Raggo, M.T., "Spyhunter: Stegspy", http://www.spy-hunter.com /stegspydownload htm. (2008).

10. Provos, N., Honeyman, P., "Hide and seek : An introduction to steganography" ***IEEE Security & Privacy Magazine***, 32- 44 (2003). http://niels.xtdnet.nl/papers/practical.pdf (2003).

11. Şahin, A., Buluş, E., Sakallı, M.T., "24-Bit Renkli Resimler Üzerinde En Önemsiz Bite Ekleme Yöntemini Kullanarak Bilgi Gizleme", ***Trakya Üniversitesi Fen Bilimleri Dergisi***,22030, Edirne-Türkiye, 17-22 (2006).

12. Mallat, S.G., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", ***IEEE.Transactions on Pattern Analysis and Machine Intelligence***, 674-693 (1989).

13. Cvejic, N., "Algorithms for Audio Watermarking and Steganography", in Department of Electrical and Information Engineering, ***Information Processing Laboratory,*** University of Oulu, Finland (2004). http://www.mp3-tech.org/programmer/docs/isbn9514273842.pdf (2004)

14. Pal, S. K., Saxena, P. K., Muttoo, S. K., "The Future of Audio Steganography", ***presented at Pacific Rim Workshop on Digital Steganography*** (STEG'02), 12-14 (2002).

15. Kim, Y.S., "Information Hiding System StegoWaveK for Improving Capacity", ***ISPA 2003,*** 271-298 (2003).

16. Farid, H., "Detecting hidden messages using higher-order statistical models", ***International Conference on Image Processing***. http://www.ists.dartmouth.edu/library/36.pdf (2002)

17. Böhme, R., Westfeld, A., "Statistical Characterisation of MP3 Encoders for Steganalysis", ***the Multimedia and Security Workshop,*** Magdeburg, Germany, 25-34 (2004).

18. Özer, H., Avcıbaş, I., Sankur, B., Memon, N., "Steganalysis of audio based on audio quality metrics", ***Security and Watermarking of Multimedia Contents V,*** Santa Clara, CA, US, 55-66 (2003).

19. Johnson, M., Lyu,S., Farid, H., "Steganalysis in recorded speech", ***Security, Steganography, and Watermarking of Multimedia Contents VII,*** San Jose, CA, US, 664-672 (2005).

20. İnternet: Hetzl, S., "Steghide," http://steghide.sourceforge.net/ (2003).

21. Ogihara, T., Nakamura, D., Yokoya, N., "Data Embedding into Pictorial Images with Less Distortion Using Discrete Cosine Transform", ***Proceedings of the 13th International Conference***, Vienna, Austria , 675-679 (1996).

22. Şahin, A., Buluş, E., Sakallı, M.T., Buluş. H.N., "Resim İçerisindeki Gizli Bilginin RQP Steganaliz Yöntemiyle Sezilmesi", ***Trakya Üniversitesi Fen Bilimleri Dergisi, 22030,*** Edirne-Türkiye. http://ab.org.tr/ab07/bildiri/63.pdf (2007)

23. Atıcı, M.A., "Steganografik Yaklaşımların İncelenmesi, Tasarımı Ve Geliştirilmesi", Yüksek Lisans Tezi, ***Gazi Üniversitesi Fen Bilimleri Enistitüsü***. Ankara, 23-24, 44-47 (2007).

24. Kessler, G.C., "An Overview of Steganography for the Computer Forensics Examiner", ***Forensic Science Communications*** 6 (3): College Burlington, Vermont.

http://www.fbi.gov/hq/lab/fsc/backissu/july2004/research/2004_03_research01.htm (2004).

25. Wang, H., Wang, S., " Cyber warfare : steganography vs. steganalysis ", ***Communications of the ACM,*** Volume 47 , Issue 10**,** New York, NY, USA, 76 - 82 (2004).

26. Pitas, I., "A Method for Signature Casting on Digital Images", ***IEEE International Conference on Image Processing***, 3: 215-218 (1999).

27. İnternet: Trithemius, J., "Steganographia:hoe est ars per occultam scripturam animi sui voluntatem absentibus aperiendi certa", www.esotericarchives.com/tritheim/stegano.htm (1997).

28. İnternet: Kolata, G., "A Mystery Unraveled, Twice", http://cryptome.unicast.org/cryptome022401/tri.crack.htm (1998).

29. İnterent: "Classical Steganography, Cardano Grille", http://library.thinkquest.org/27993/crypto/steg/classic1.shtml (1996).

30. İnternet: Blue, P., Pages, W., "What the Returning POWs Said About Missing Men", www.miafacts.org/pages.htm (2000).

31. İnterent: XRCE TeXnology Showroom "Digital watermarking in printed images: Stochastic halftone screen design", Fact Sheet, http://www.xrce.xerox.com/showroom/fs/stochas.pdf (2002).

32. İnterent: Gafford, R., "The Operational Potential of Subliminal Perception", https://www.cia.gov/library/center-for-the-study-of-intelligence/ kent - csi / vol2 no2/html/v02i2a07p_0001.htm (1995).

33. Sayood, K., "Introduction to Data Compression", ***Morgan Kauffman Publishers, Inc. 340 Pine Street,*** Sixth Floor, San Francisco, CA 94104-3205, USA, 309-333 (1996).

34. Preneel, B., Rijmen, V., Bosselaers, A., "Principles and performance of cryptographic algorithms", ***Dr. Dobb's Journal,*** 23 (12): 126-131 (1998).

35. Biryukov, A., "Methods of Cryptanalysis", Mathematics and Computer Science, ***Weizmann Institute of Science***. PHd Thesis, Germany, 29-45 (1999).

36. Stinson, D.R., "Cryptography: Theory and Practice, Second Edition", ***CRC Press,*** 22-25 (2002).

37. İnternet: "WetStone Technologies Inc.", http://www.wetstonetech.com/faq_stego.html (1997).

38. İnternet: "OutGuess - Steganography Detection", http://www.outguess.org/detection (2007).

39. Fridrich, J., Du R., Meng, L., "Steganalysis of LSB Encoding in Color Images", ***Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conferece,*** New York City, USA, 3: 1279-1282 (2000).

40. El Loco G.,"Extracting data embedded with JSteg", http://www. guillermito2. net/ stegano/jsteg/index.html (2004).

41. Özer, H., Sankur, B., Memon, N., Avcıbaş, İ., "Detection Of Audio Covert Channels Using Statistical Footprints Of Hidden Messages", ***Digital Signal Processing***, 16 (4): 389-401 (2006).

42. Lau, S., "An Analysis of Terrorist Groups' Potential Use of Electronic Steganografy", ***SANS Security Essentials GSEC Practical Assignment Version 1,*** 3-18 (2003).

43. Canbek, G., Sağıroğlu, Ş., "Bilgi ve Bilgisayar Güvenliği Casus Yazılımlar ve Korunma Yöntemleri", ***Grafiker***, Ankara, 1-50 (2006).

44. Watkins J., "Steganography - Messages Hidden in Bits**",** Multimedia Systems Coursework, ***Department of Electronics and Computer Science,*** University of Southampton, 8-10 (2001).

45. Engle S., "Current State of Steganography: Uses, Limits, Implications", University of California, Davis. www.csif.cs.ucdavis.edu/~engle/stego.pdf (2004).

46. Lou, D.C., Liu, J.L., "Steganographic Method for Secure Communications", ***Computers and Security,*** 21 (5): 449-460 (2002).

47. Yu, Y.H., Chin-Chen C., Lin, I.C., "A new steganographic method for color and grayscale image hiding", ***Computer Vision and Image Understanding***, New York, NY, USA, 107 (3): 183-194 (2006).

48. Cachin C., "An Information-Theoretic Model for Steganography", ***Lecture Notes In Computer Science***, Springer-Verlag London, UK, 1525: 306-318 (2004).

49. Grover, D., "Steganography for Identifying Ownership of Copyright", ***Computer Law and Security Report,*** 14 (2): 121-122 (1998).

50. Chandramouli, R., Memon, N.,"Analysis of LSB Based Image Steganography Techniques", ***Proceedings of the International Conference on Image Processing*****,** Thessalonica, Ekim, Yunanistan, 1019-1022 (2001).

51. Chwei-Shyong, T., Chin-Chen C., Tung-Shou C., "Sharing multiple secrets in digital images", ***Journal of Systems and Software*** 64 (2): 163-170 (2002).

52. Da-Chun, W., Wen-Hsiang T., "A steganographic method for images by pixel-value differencing", ***Pattern Recognition Letters***, Elsevier Science Inc. New York, NY, USA, 24 (9-10): 1613-1626 (2003).

53. Chin-Chen, C., Tung-Shou, C., Lou-Zo C., "A steganographic method based upon JPEG and quantization table modification", ***nformation Sciences Informatics and Computer Science: An International Journal,*** Volume 141, Issue 1-2, Elsevier Science Inc. New York, NY, USA, 123-138 (2002).

54. Marvel, L., Retter, C., "Spread Spectrum Image Steganography", ***IEEE Transactions on Image Processing***, 8(8): 1075-1083 (1999).

55. E.Beşdok, "Hiding information in multispectral spatial images", ***Department of Geodesy and Photogrammetry Engineering, Faculty of Engineering***, Erciyes University, 38039, Kayseri, Turkey, 15-24 (2003).

56. İnternet:Bloisi, D., Iocchi, L., "Image Based Steganography and Cryptography" http://www.dis.uniroma1.it/~iocchi/publications/iocchi-visapp07-isc.pdf (2004).

57. İnternet: Potdara,V.M., Khana, M.A., Changa, E., Ulierub, M., Worthington, P.R., "E-Forensics Steganography System For Secret Information Retrieval". http//espace. lise. curtin. edu. au / archive / 00000677 / 01 / 442 e-Forensics steganography systems for secret information r.pdf (2005).

58. Westfeld, A., Pfitzmann, A., "Attacks on Steganographic Systems: Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and S-Tools–and Some Lessons Learned", 3rd International Workshop on Information Hiding. http://www.ece.cmu.edu/~adrian/487-s06/westfeld-pfitzmann-ihw99.pdf (2000).

59. Avcıbaş, İ., "Audio Steganaliz With Content Independent Distortion Measures", ***IEEE Signal Processing Letters***, 13(2): 92-96 (2006).

60. İnternet: Wang, Y., Moulin, P., "Steganalysis of Block-DCT Image Steganography", Proc. ***IEEE Workshop on Statistical Signal Processing***. http://www.ifp.uiuc.edu/~moulin/Papers/ssp03-ying.pdf (2003).

61. Johnson, N.F., Jajodia, S., "Steganalysis of Images Created Using Current Steganography Software", ***Lecture Notes in Computer Science***, 1525: 273-289 (1998).

62. İnternet: Simmons, G.J., "The prisoner's problem and the subliminal channel. Advances in Cryptology", ***Proceedings of CRYPTO 83***. http://dsns.csie.nctu.edu.tw/research/crypto/HTML/PDF/C83/51.PDF (1983).

63. İnternet: Raggo, M.T., "Identifying and cracking steganography programs", Black Hat 2004 conference. http://www.blackhat.com/html/bh-media-archives/bh-archives-2004.html(2004).

64. İnternet: Russ Rogers. The keys to the kingdom. Black hat Japan, http://blackhat.com/presentations/bh-asia-04/bh-jp-04-pdfs/bh-jp-04-rogers.pdf. (2004).

65. Bender, W., Gruhl, D., Morimoto, N., Lu, A., "Techniques for data hiding", ***IBM Syst. J***. 35, (3&4), 313-336. (1996).

66. Caldwell, 2nd Lt. J., "Seganography", ***CROSSTALK the Jornal of Defense Software Engineering,*** 25-27 (2003).

67. Internet : Chmura, J., Haller, B., "Steganography", EECS 600-2004. http://www.jchmura.com/presentations/jlc_bah_steganography.pdf (2004).

68. Kipper, G., "Investigator's guide to steganography, first edition", ***CRC Press*** ,Auerbach publications, 99-146, (2003).

69. İnternet: Fridrich, J., Goljan, M., Soukal, D., "Searching for the stego key", ***Proceedings of SPIE: Security,*** Steganography, and Watermarking of Mul104 timedia Contents VI, 5306. http://www.ijde.org/docs/IJDE-LeiglandKrings.pdf (2004).

70. Fridrich, J., Goljan, M., Du, R., "Steganalysis based on jpeg compatibility", Special session on Theoretical and Practical Issues in Digital Watermarking and Data Hiding***, SPIE Multimedia Systems and Applications IV.*** http://www.ws.binghamton.edu/fridrich/Research/jpgstego01.pdf (2001).

71. Ru et al., "Audio Steganalysis Based On'Negative Resonance Phenomenon' Caused By Steganographic Tools.", ***Journal of Zhejiang University Science A***, 7(4): 577-583 (2006).

72. Westfeld, A., Pfitzmann, A., "Attacks on steganographic systems," Information Hiding. Third International Workshop. http://os.inf.tu dresden.de/˜westfeld/publikationen/ihw99.pdf. (1999).

73. İnternet: Hunt, W. L., "PhotoTiled Pictures Homepage", http:// home. earthlink.net/~wlhunt/ (2006).

74. İnternet: http://www.prismaticsoftware.com/Phototile/PhotoTile.html (2006).

75. İnternet: http://www.pctitcoJas.nut/fabien/stegano.graphy/mp3slego/ (2002).

76. İnternet: https: //www. spammimic.com/ (2007).

77. Kessler, G.C., "An Overview of Steganography for the Computer Forensics Examiner", ***Forensic Science Communications*** 6 (3): 1026-1029 (2004).

78. İnternet: http://www.stegoarchive.com (2005).

79. İnternet: http://homepages.tu-darmstadt.de/~rkibria/ adresinden (2000).

80. İnternet: http://www.tiac.net/users/korejwa/jsteg.htm (2003).

81. İnternet: http://www.darkside.com.au/GIFshuffle/ (2003).

82. İnternet: http://www.neobytesolutions.com/invsecr (2004).

83. İnternet: Young, B., "Digital Piracy: Theft in the Modern Age", http://biz.howstuffworks.com/by-digital-piracy.htm?printable=1 (2008).

84. İnternet: Lathrop, D. A. Captain, USAF, "Viral Computer Warfare Via Activation Engine Employing Steganography", http://www.iwar.org.uk/iwar/resources/usaf/maxwell/students/2000/afit-gcs-eng-00m-14.htm (2000).

85. Fridrich, J., Goljan, M., Du, R., "Detecting LSB steganography in color, and gray-scale images", ***Multimedia, IEEE*** **,** 8 (4): 22-28, (2001).

86. Fridrich, J., Goljan, M., Du, R., "Reliable Detection of LSB Steganography in Color and Grayscale images", Proc. of the ACM ***Workshop on Multimedia and Security,*** 27-30 (2001).

87. Farid, H. "Detecting Steganographic Messages in Digital images." ***Technical Report TR2001-412, Dartmouth College,*** 1-9 (2001).

88. İnternet: U.S. Department of Justice , "Electronic Crime Scene Investigation, A Guide for First Responders", http://www.iwar.org.uk / ecoespionage / resources / cybercrime / ecrime-scene investigation.pdf (2001).

89. Fridrich, J., "Minimizing the embedding impact in steganography", ***Proceeding of the 8th Workshop on Multimedia and Security,*** Geneva-Switzerland, 2-10, (2006).

90. Avcıbaş, İ., Kharrazi, M., Memon, N., Sankur,B., "Image Steganalysis With Binary Similarity Measures," ***EURASIP Journal on Applied Signal Processing***, (17): 2749- 2757 (2005).

91. Sankur B., Memon, N., Avcıbaş, İ., "Automatic Detection of the Presence of Stegosignals and Watermarks in Images," ***European Conference on***

Visual Perception, 38-38 (2001).

92. Fridrich, J., Goljan, M., "Practical steganalysis of digilal images state of the art", ***Proc. SPIE Photonics West,*** 4675: 1-13 (2002).

93. İnternet: http://www.dc3.mil/dcci/catalog.htm (2007).

94. İnternet: http://www.cybersciencelab.com (2007).

95. Stephen, K., Thompson, J.D., " Performance Analysis of a New Semiorthogonal Spline Wavelet Compression Algorithm for Tonal Medical Images", ***Medical Physics***, 27 (2): 276-288 (2000).

96. Katsenbeisser, S., Petitcolas, F., "Information hiding Techniques for Steganography and Digital Watermarking", ***Artech House***, Boston. 12-29 (2000).

97. Pickholtz, R. L., Schilling, D. L. and Milstein, L. B., "Theory of Spread-Spectrum Communications-A Tutorial", ***IEEE Transactions on Communications***, 30(5): 855–884 (1982).

98. Sağıroğlu, Ş.. Tunçkanat, M., "A Secure Internet Communication Tool", ***Turkish Journal of Telecommunications,*** 1 (1): 40-46 (2002).

99. Smirnov, M.V., "Holographic approach to embedding hidden watermarks in a photographic image", ***Journal of Optical Technology,*** 72 (6): 464-468 (2002).

100. İnternet: http://www.marie-stuart.co.uk/ (2008).

101. Adli, A., Nakao, Z., "Three steganography algorithms for MIDI files", ***Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference***, Guangzhou, China, 4: 2401-2404, 18-21 (2005).

102. Sui, X.G., Luo, H., "A new steganography method based on hypertext", Radio Science Conference, ***2004. Proceedings. 2004 Asia-Pacific,*** 18-184, 24-27 (2004).

103. İnternet: http://www.w3schools.com/xml/xml_whatis.asp (2008).

104. El-Khalil,R., Keromytis, A.D., "Hydan: Hiding Information in Program Binaries" http://www.cs.columbia.edu/~angelos/Papers/hydan.pdf (2008).

105. Center for Innovative Computer Applications, "Image File Formats List", http://www.cica.indiana.edu/graphics/image.formats.html (2008).

106. İnternet: Astrowsky, B. H., "Steganography: Hidden Images, A New Challenge in the Fight Against Child Porn", http://www.antichildporn.org/steganog.html (2008).

107. İnternet: The High Technology Crime Advisory Committee "High Technology Crime in California", http://www.ocjp.ca.gov/publications/pub_htk1.pdf (2008).

108. Cochtan, J.T., Captain, USAF, "Steganographic Computer Warfare", http://www.iwar.org.uk/iwar/resources/usaf/maxwell/students/2000/afit-gcseng-00m-03.htm (2008).

109. İnternet: Howstuffworks.com "How Anti-shoplifting Devices Work", http://www.howstuffworks.com/anti-shoplifting-device.htm/printable (2008).

110. İnternet: Smithsonian American Art Museum ""Hidden" Letters", http://nmaa-ryder.si.edu/education/kids/cappy/14cactivity2.html (2008).

111. Proves, N., Honeyman, P., "Detecting Steganographic content on the internet", ***Tech. Rep. CITI 01-1a, University of Michigan,*** 1-14 (2001).

112. Jamil, T., "Steganography: the art of hiding information in plain sight", ***Potentials, IEEE,*** 18 (1): 10-12 (1999).

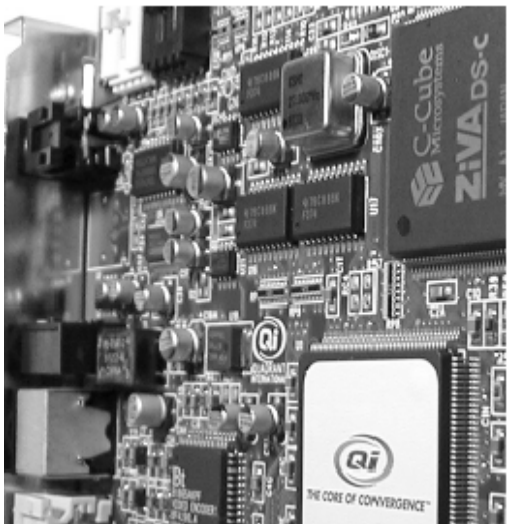
EKLER

EK-1 İmajlar

Im001.png

Im004.png

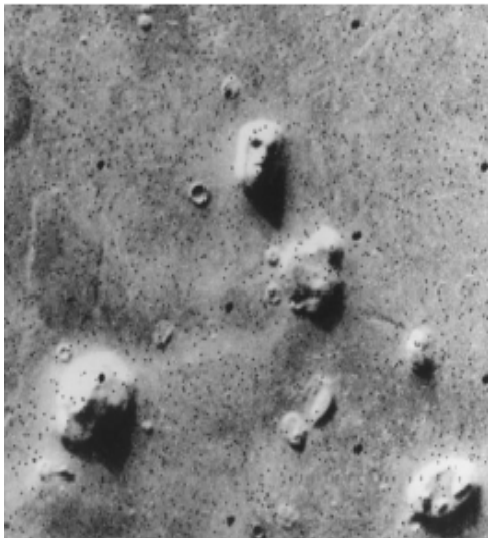
Im002.png



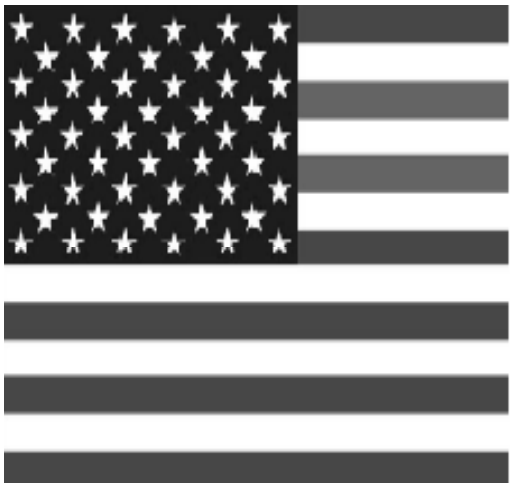
Im006.png

EK-1 (Devam) İmajlar

Im003.png

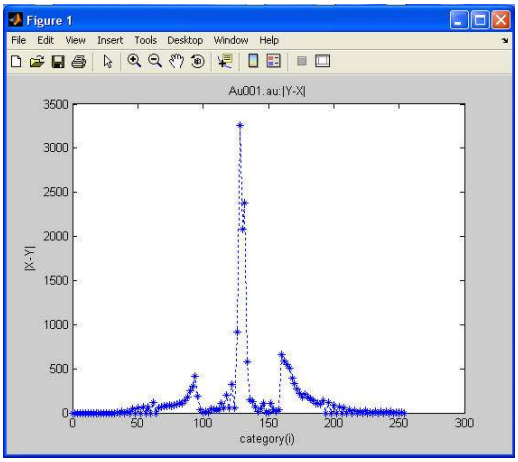


Im007.png

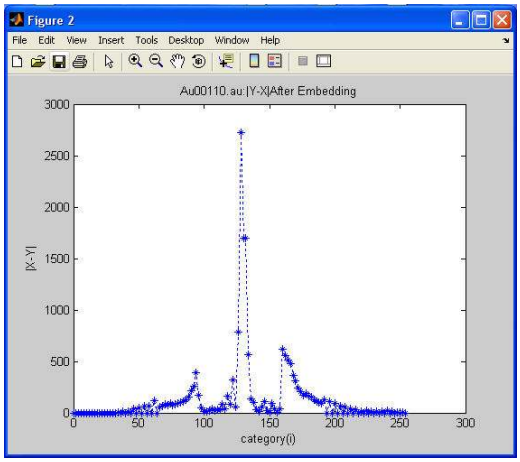


Im005.png

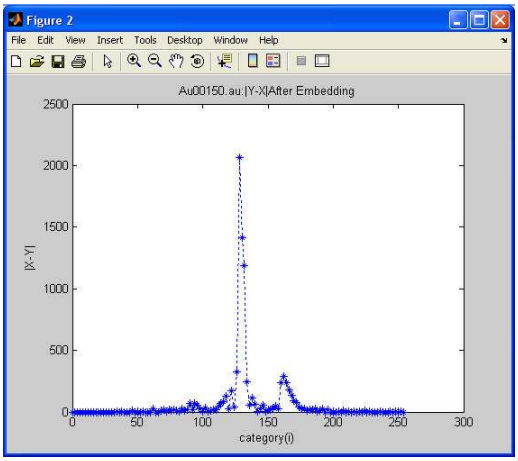
EK-2 Ses Dosyaların Görüntüsü



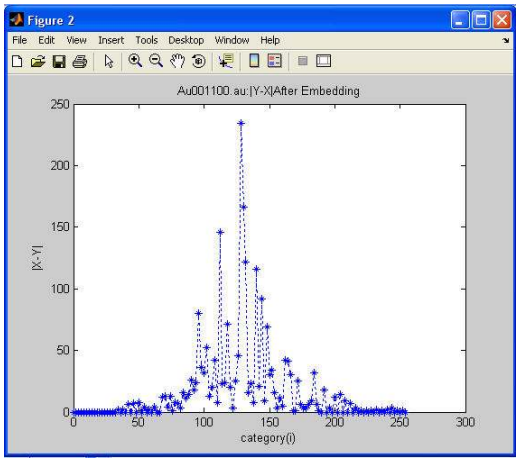
(Au001)Orijinal Sese Ait Audio Unit Histegramı (%0 veri gizlenmiş)



(Au00110)Stego Sese Ait Audio Unit Histegramı (%10 veri gizlenmiş)

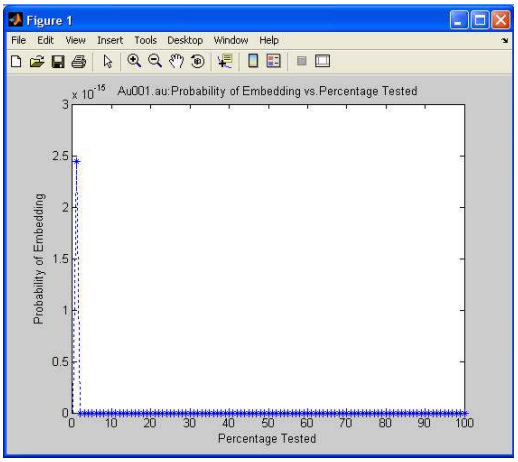


(Au00150)Stego Sese Ait Audio Unit Histegramı (%50 veri gizlenmiş)

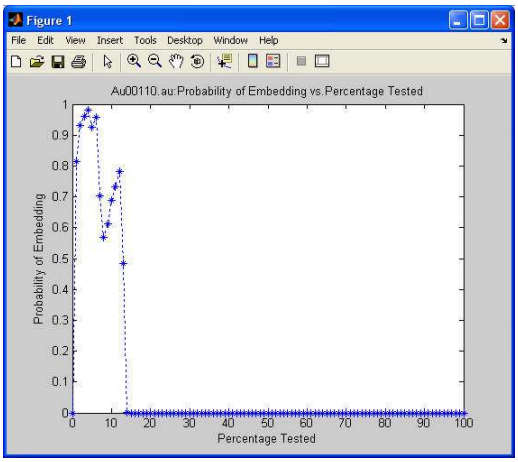


(Au001100)Stego Sese Ait Audio Unit Histegramı (%100 veri gizlenmiş)

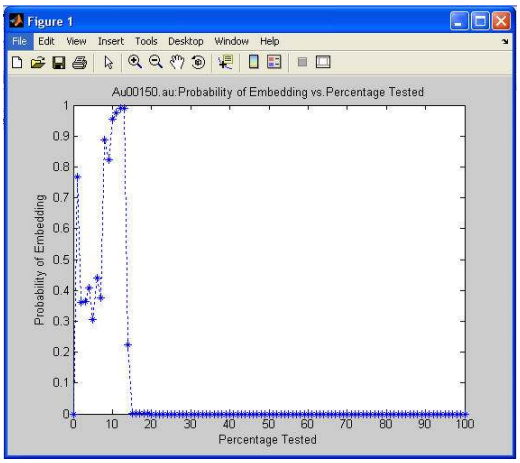
EK-2 (Devam) Ses Dosyaların Görüntüsü



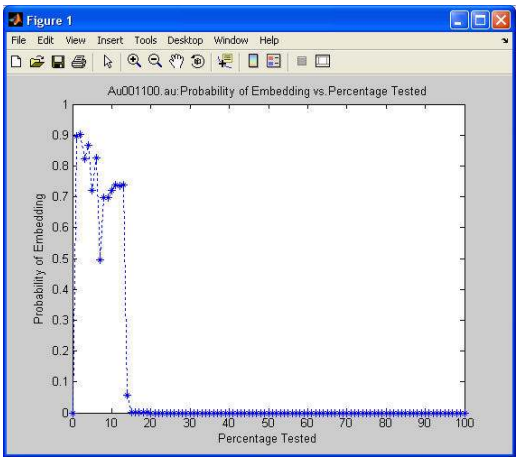
(Au001)Orijinal Sese Ait Audio Unitin Analizi (%0 veri gizlenmiş)



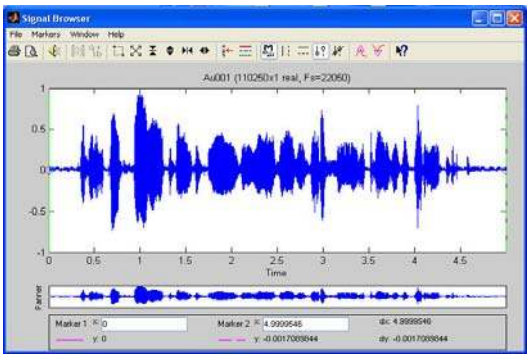
(Au00110)Stego Sese Ait Audio Unitin Analizi (%10 veri gizlenmiş)



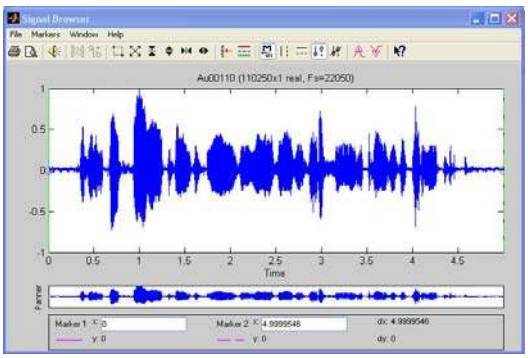
(Au00150)Stego Sese Ait Audio Unitin Analizi (%50 veri gizlenmiş)



(Au001100)Stego Sese Ait Audio Unitin Analizi (%100 veri gizlenmiş)

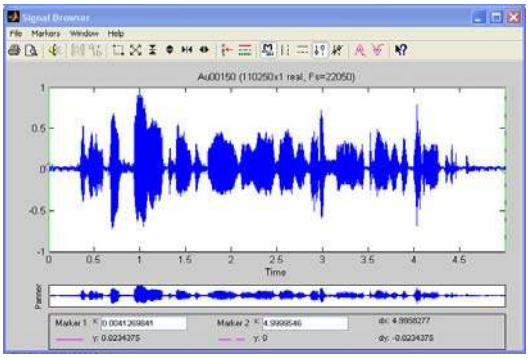


(Au001)Orijinal Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%0verigizlenmiş)

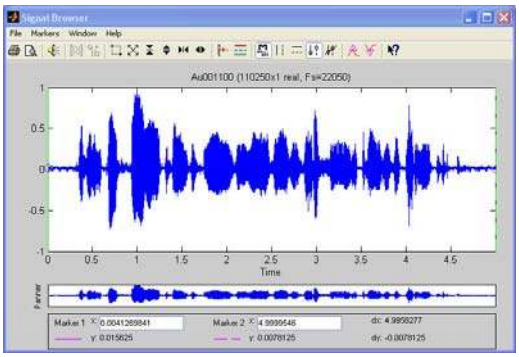


(Au00110)Stego Sese Ait Audio Unitin İşaretÖrünTarayıcısı(%10veri gizlenmiş)

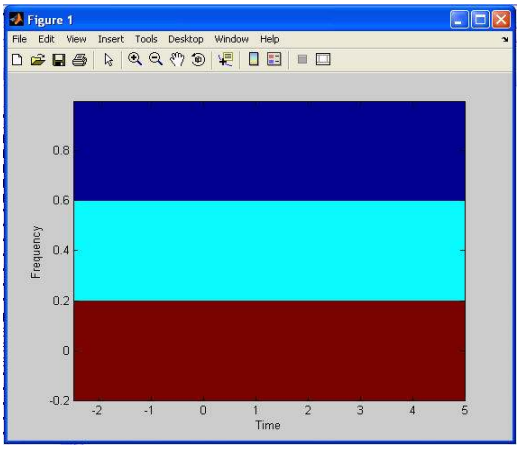
EK-2 (Devam)Ses Dosyaların Görüntüsü



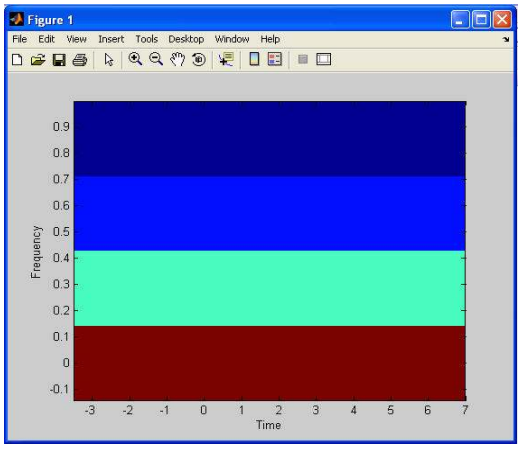
(Au00150)Stego Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%50 veri gizlenmiş)



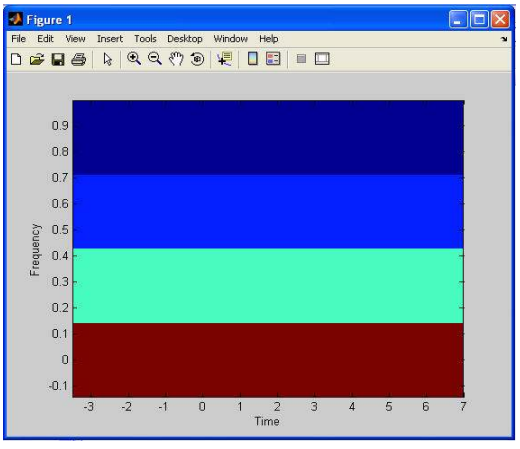
(Au001100)Stego Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%100 veri gizlenmiş)



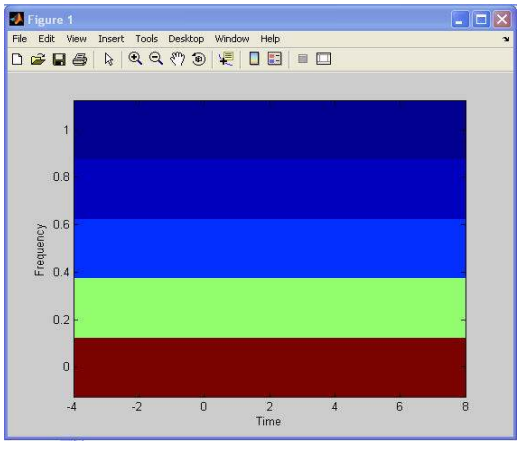
(Au001)Orijinal Sese Ait Audio Unitin Spektrumu (%0 veri gizlenmiş)



(Au00110)Stego Sese Ait Audio Unitin Spektrumu (%10 veri gizlenmiş)

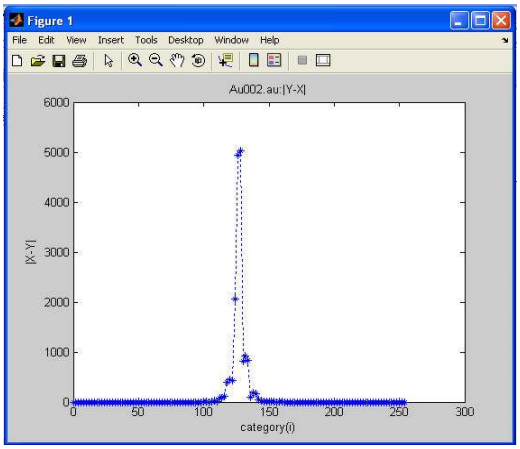


(Au00150)Stego Sese Ait Audio Unitin Spektrumu (%50 veri gizlenmiş)

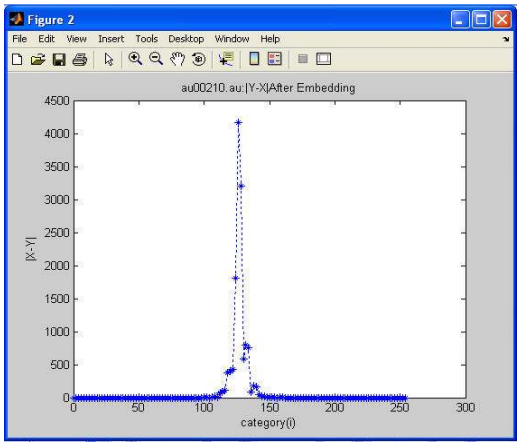


(Au001100)Stego Sese Ait Audio Unitin Spektrumu (%100 veri gizlenmiş)

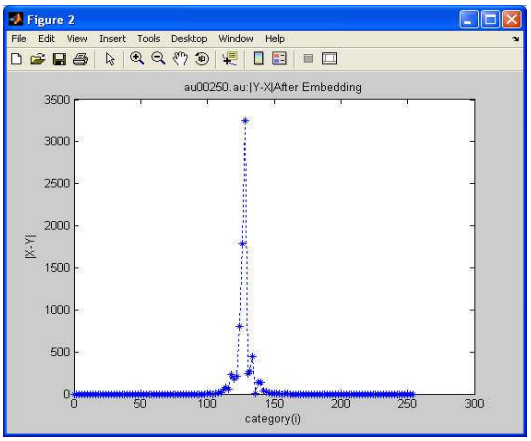
EK-2 (Devam)Ses Dosyaların Görüntüsü



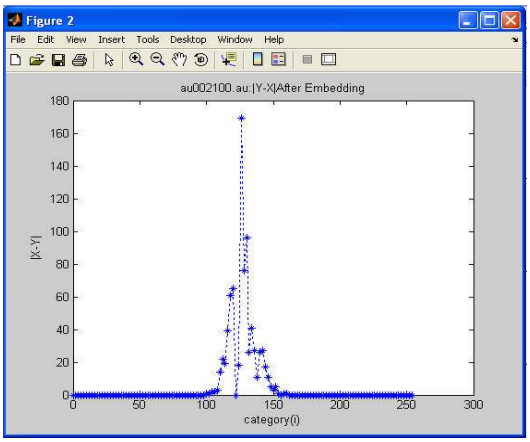
(Au002)Orijinal Sese Ait Audio Unit Histegramı (%0 veri gizlenmiş)



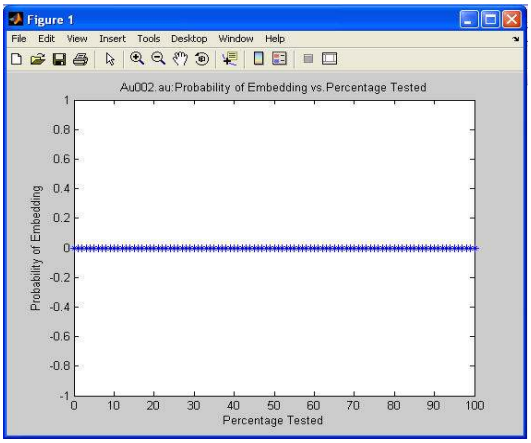
(Au00210)Stego Sese Ait Audio Unit Histegramı (%10 veri gizlenmiş)



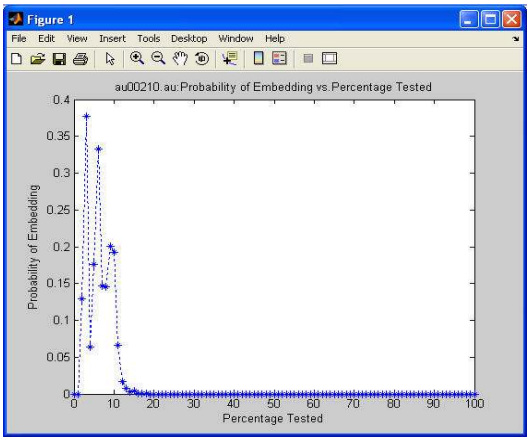
(Au00250)Stego Sese Ait Audio Unit Histegramı (%50 veri gizlenmiş)



(Au002100)Stego Sese Ait Audio Unit Histegramı (%100 veri gizlenmiş)

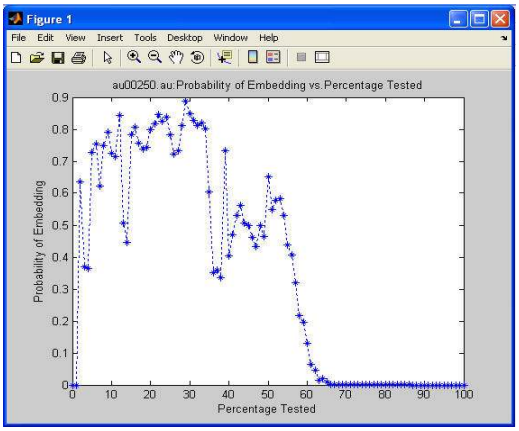


(Au002)Orijinal Sese Ait Audio Unitin Analizi (%0 veri gizlenmiş)

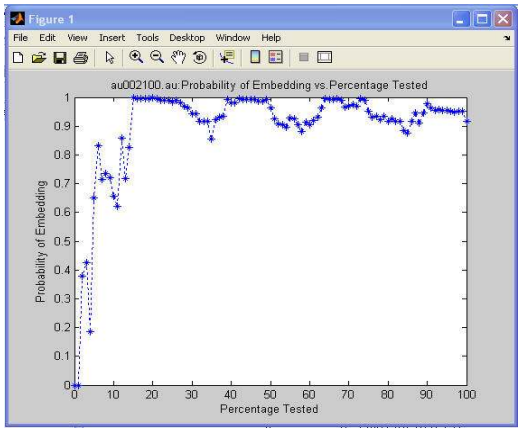


(Au00210)Stego Sese Ait Audio Unitin Analizi (%10 veri gizlenmiş)

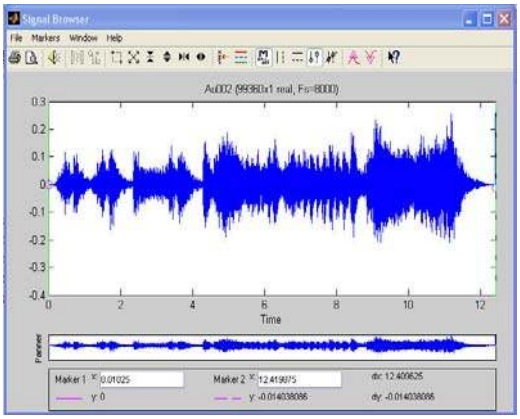
EK-2 (Devam)Ses Dosyaların Görüntüsü



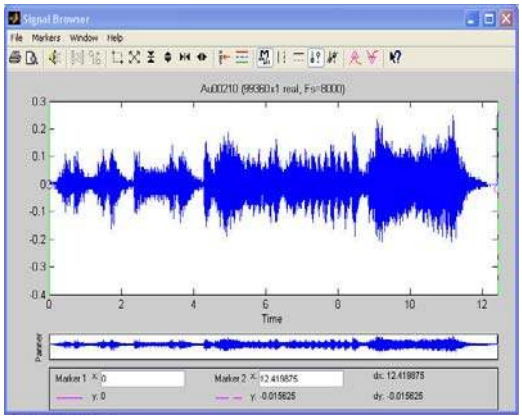
(Au00250)Stego Sese Ait Audio Unitin Analizi (%50 veri gizlenmiş)



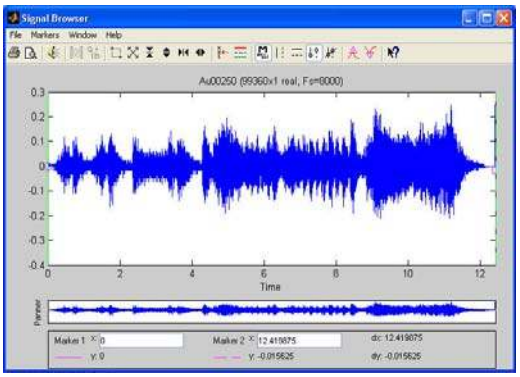
(Au002100)Stego Sese Ait Audio Unitin Analizi (%100 veri gizlenmiş)



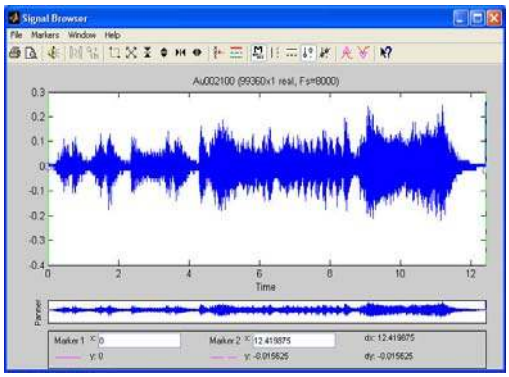
(Au001)Orijinal Sese Ait Audio Unitin İşaret Örün Tarayıcısı(%0 veri gizlenmiş)



(Au00110)Stego Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%10 veri gizlenmiş)

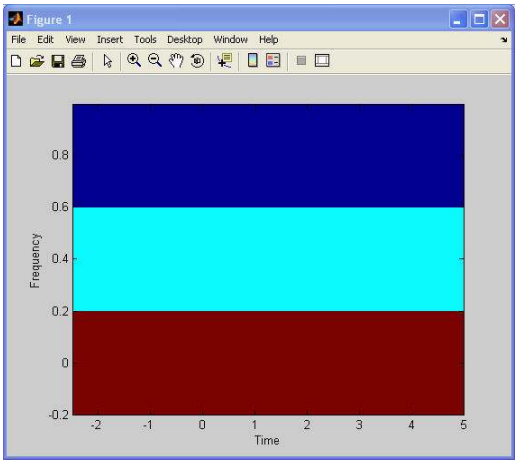


(Au00250)Stego Sese Ait Audio Unitin İşaretÖrün Tarayıcısı(%50verigizlenmiş)

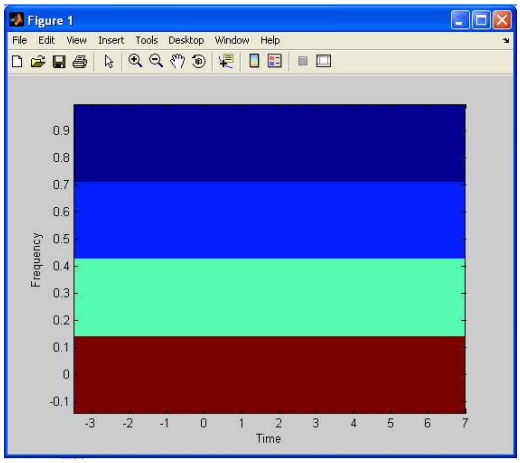


(Au002100)StegoSeseAit AudioUnitin İşaretÖrünTarayıcısı(%100verigizlenmiş)

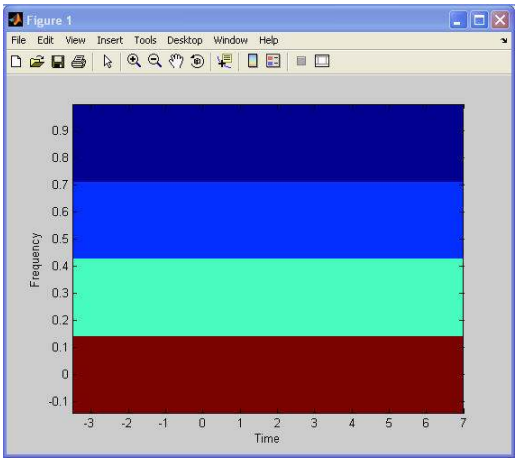
EK-2 (Devam)Ses Dosyaların Görüntüsü



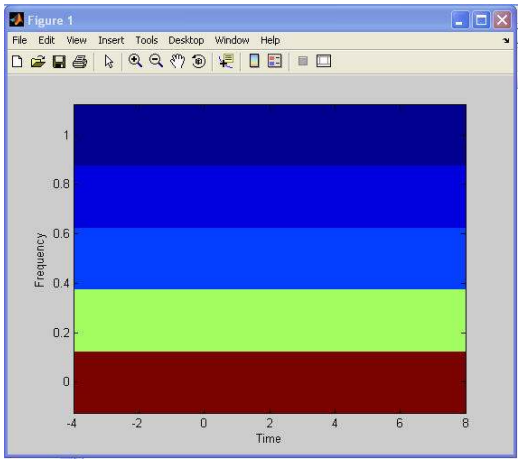
(Au002)Orijinal Sese Ait Audio Unitin Spektrumu (%0 veri gizlenmiş)



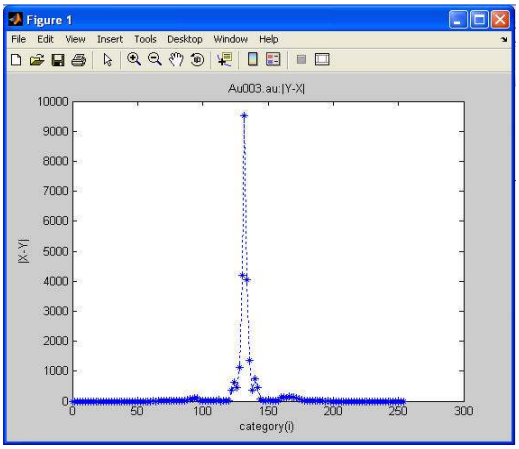
(Au00210)Stego Sese Ait Audio Unitin Spektrumu (%10 veri gizlenmiş)



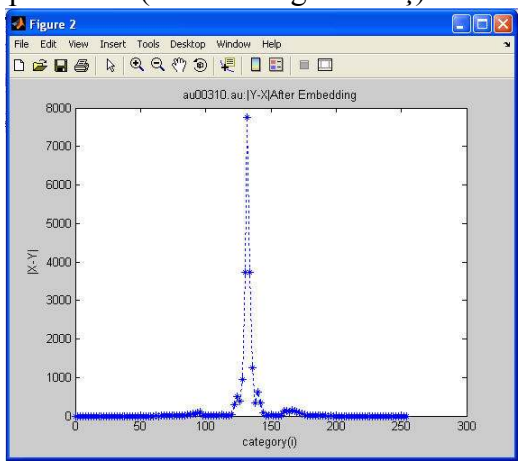
(Au00250)Stego Sese Ait Audio Unitin Spektrumu (%50 veri gizlenmiş)



(Au002100)Stego Sese Ait Audio Unitin Spektrumu (%100 veri gizlenmiş)

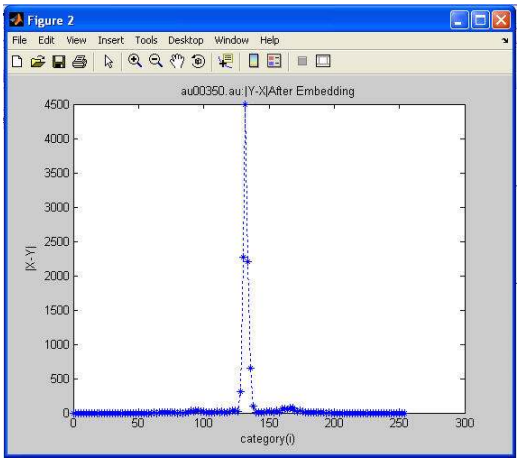


(Au003)Orijinal Sese Ait Audio Unit Histegramı (%0 veri gizlenmiş)

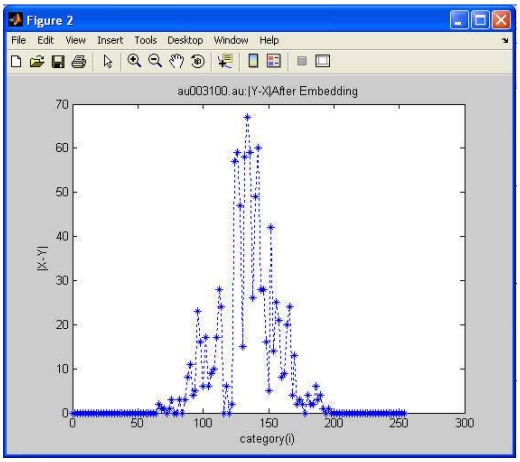


(Au00310)Stego Sese Ait Audio Unit Histegramı (%10 veri gizlenmiş)

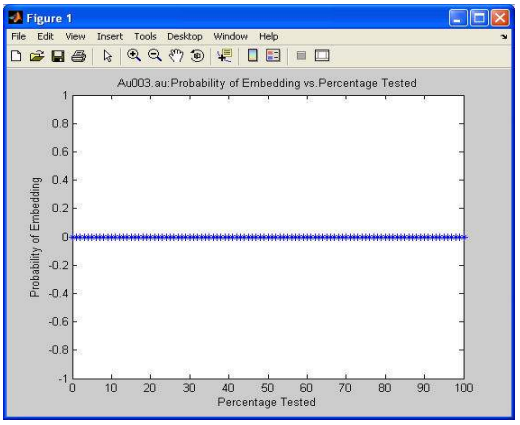
EK-2 (Devam)Ses Dosyaların Görüntüsü



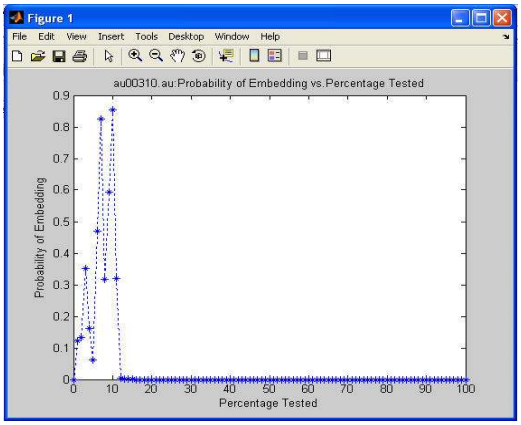
(Au00350)Stego Sese Ait Audio Unit Histegramı (%50 veri gizlenmiş)



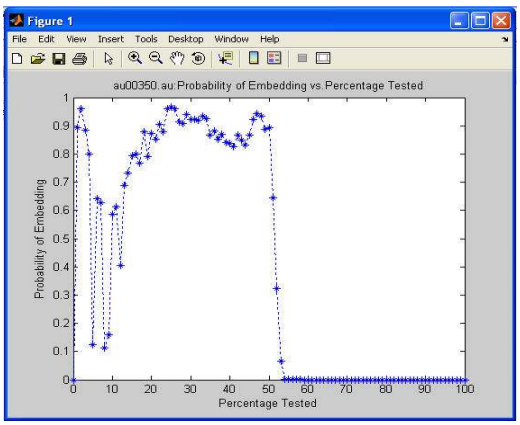
(Au003100)Stego Sese Ait Audio Unit Histegramı (%100 veri gizlenmiş)



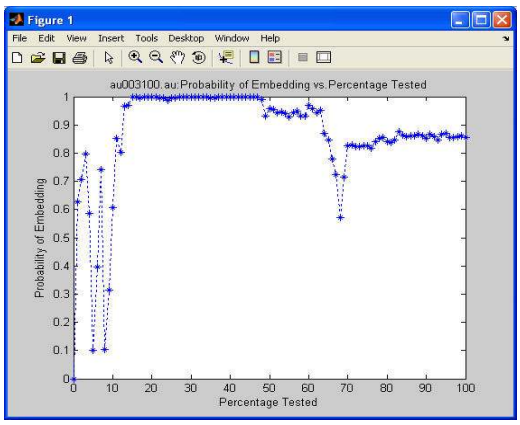
(Au003)Orijinal Sese Ait Audio Unitin Analizi (%0 veri gizlenmiş)



(Au00310)Stego Sese Ait Audio Unitin Analizi (%10 veri gizlenmiş)

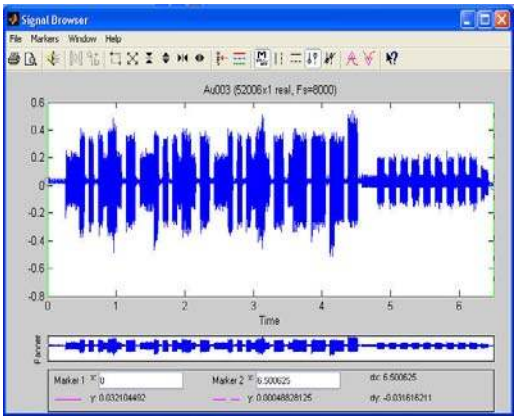


(Au00350)Stego Sese Ait Audio Unitin Analizi (%50 veri gizlenmiş)

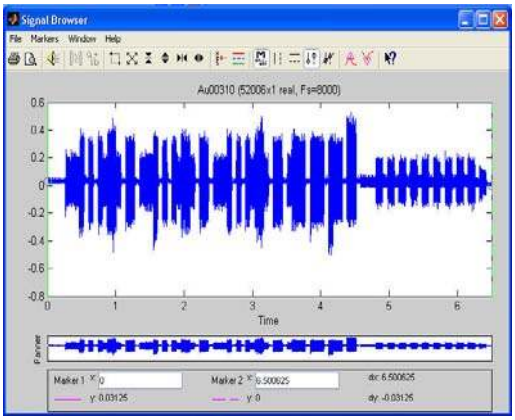


(Au003100)Stego Sese Ait Audio Unitin Analizi (%100 veri gizlenmiş)

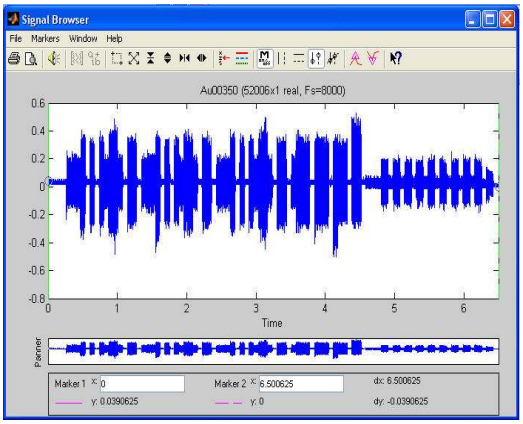
EK-2 (Devam)Ses Dosyaların Görüntüsü



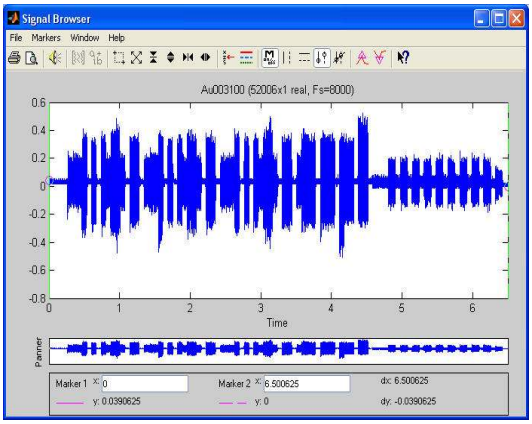
(Au003)Orijinal Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%0 veri gizlenmiş)



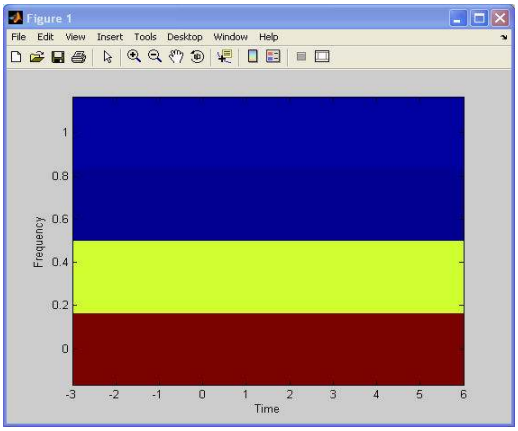
(Au00310)Stego Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%10 veri gizlenmiş)



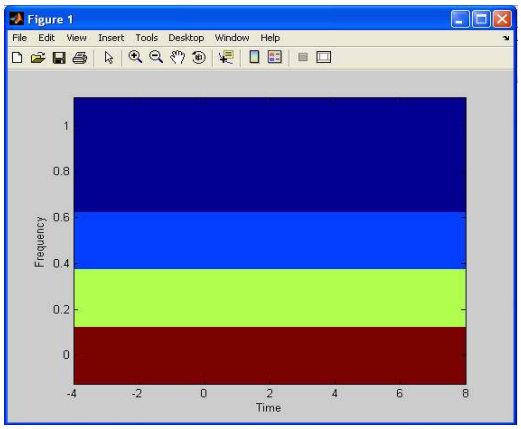
(Au00350)Stego Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%50 veri gizlenmiş)



(Au003100)Stego Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%100 veri gizlenmiş)

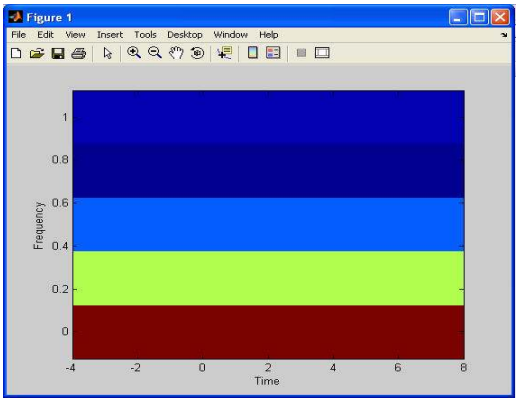


(Au003)Orijinal Sese Ait Audio Unitin Spektrumu (%0 veri gizlenmiş)

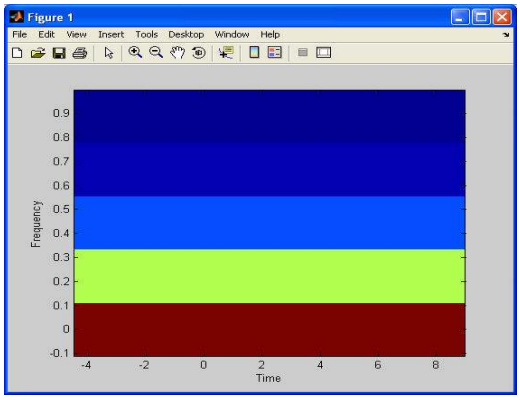


(Au00310)Stego Sese Ait Audio Unitin Spektrumu (%10 veri gizlenmiş)

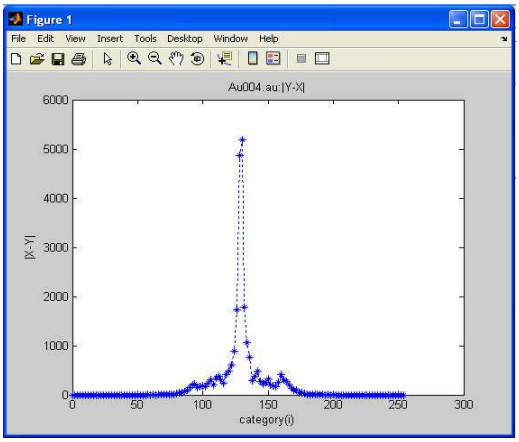
EK-2 (Devam)Ses Dosyaların Görüntüsü



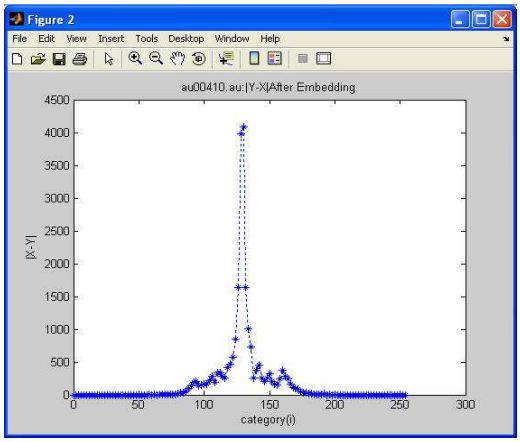
(Au00350)Stego Sese Ait Audio Unitin Spektrumu (%50 veri gizlenmiş)



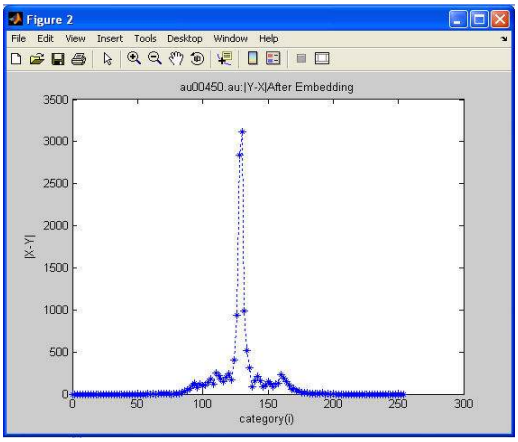
(Au003100)Stego Sese Ait Audio Unitin Spektrumu (%100 veri gizlenmiş)



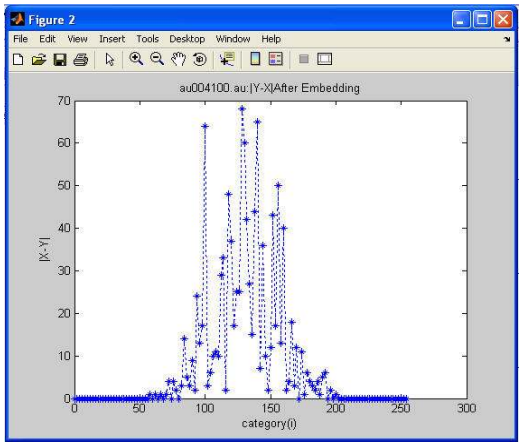
(Au004)Orijinal Sese Ait Audio Unit Histegramı (%0 veri gizlenmiş)



(Au00410)Stego Sese Ait Audio Unit Histegramı (%10 veri gizlenmiş)

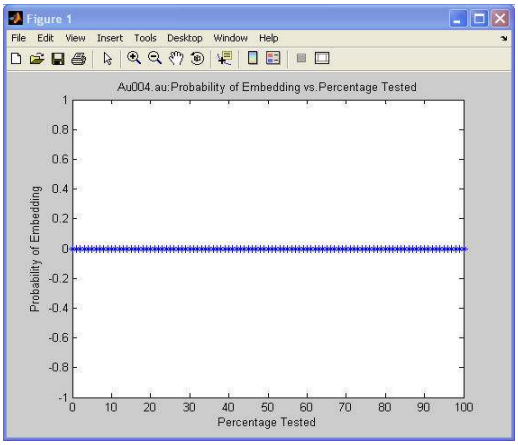


(Au00450)Stego Sese Ait Audio Unit Histegramı (%10 veri gizlenmiş)

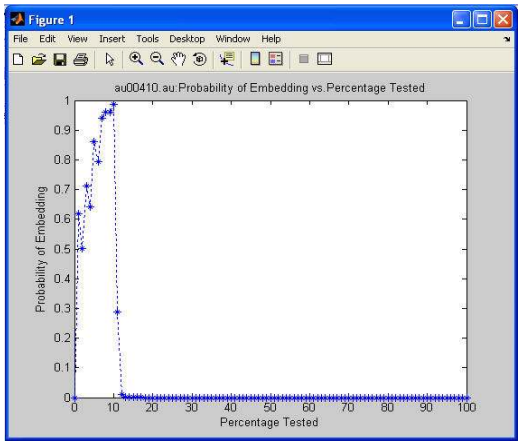


(Au004100)Stego Sese Ait Audio Unit Histegramı (%100 veri gizlenmiş)

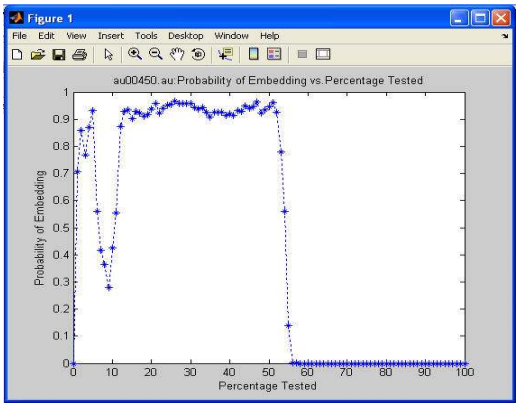
EK-2 (Devam)Ses Dosyaların Görüntüsü



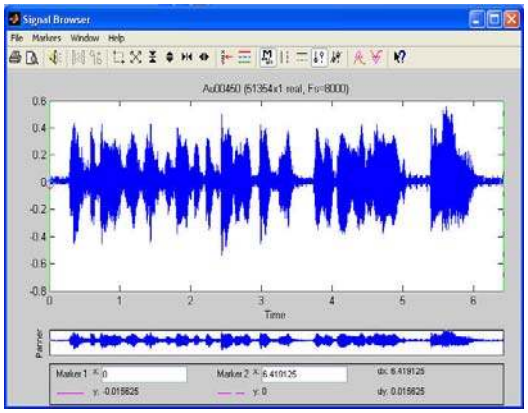
(Au004)Orijinal Sese Ait Audio Unitin Analizi (%0 veri gizlenmiş)



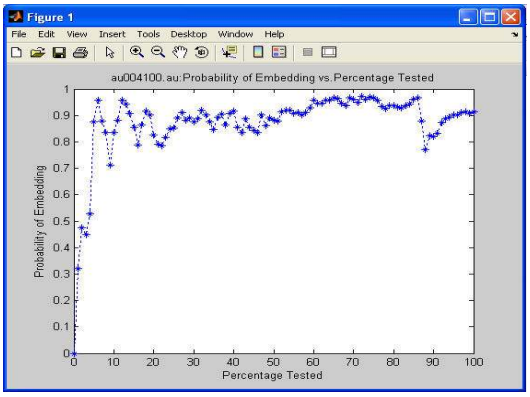
(Au00410)Stego Sese Ait Audio Unitin Analizi (%10 veri gizlenmiş)



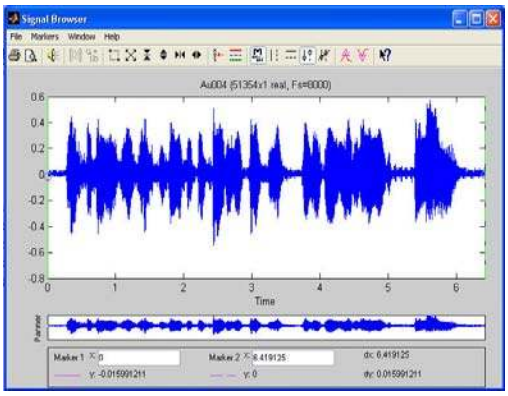
(Au00450)Stego Sese Ait Audio Unitin Analizi (%50 veri gizlenmiş)



(Au00450)Stego Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%50 veri gizlenmiş)

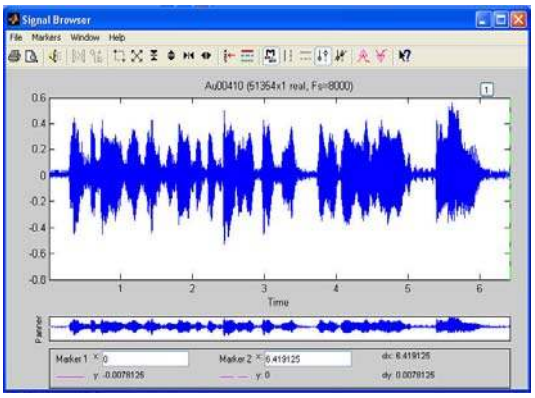


(Au004100)Stego Sese Ait Audio Unitin Analizi (%100 veri gizlenmiş)

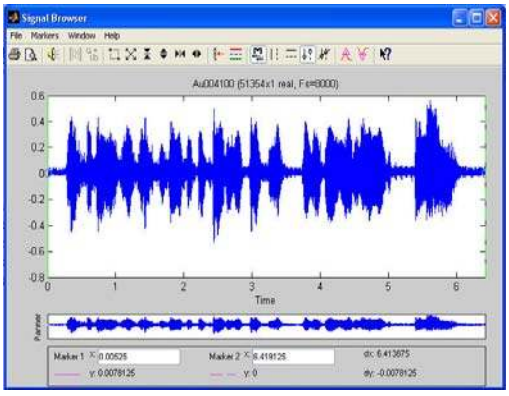


(Au004)Orijinal Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%0 veri gizlenmiş)

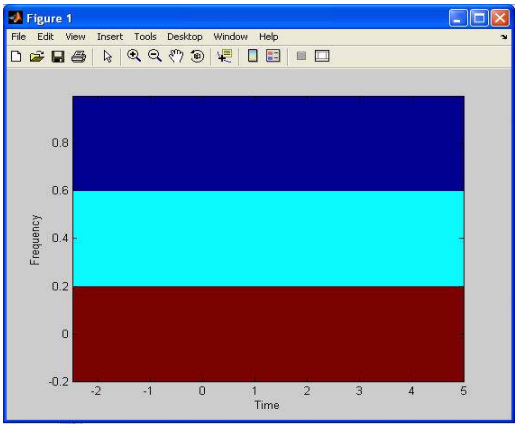
EK-2 (Devam)Ses Dosyaların Görüntü



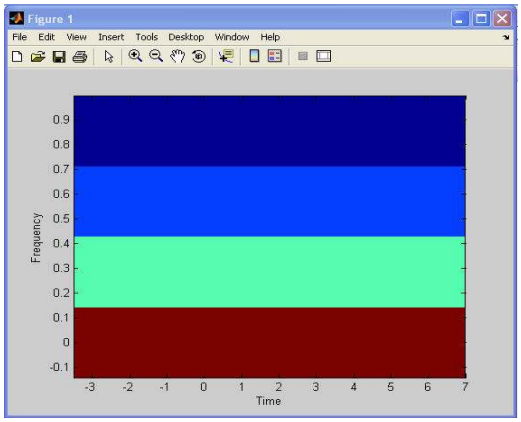
(Au00410)Stego Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%10 veri gizlenmiş)



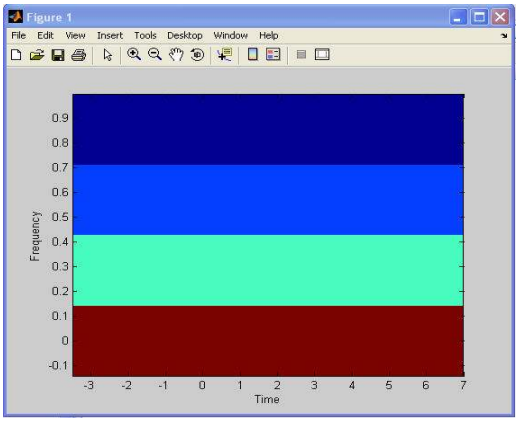
(Au004100)Stego Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%100 veri gizlenmiş)



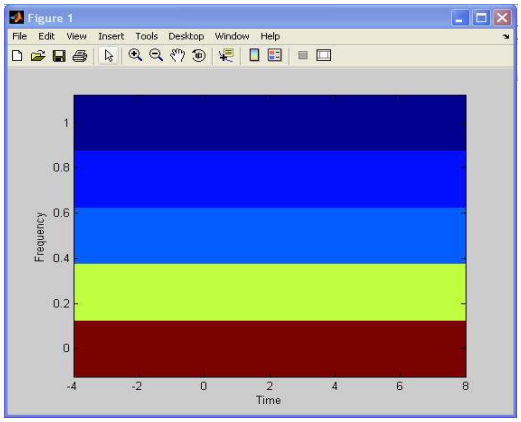
(Au004)Orijinal Sese Ait Audio Unitin Spektrumu (%0 veri gizlenmiş)



(Au00410)Stego Sese Ait Audio Unitin Spektrumu (%100veri gizlenmiş)

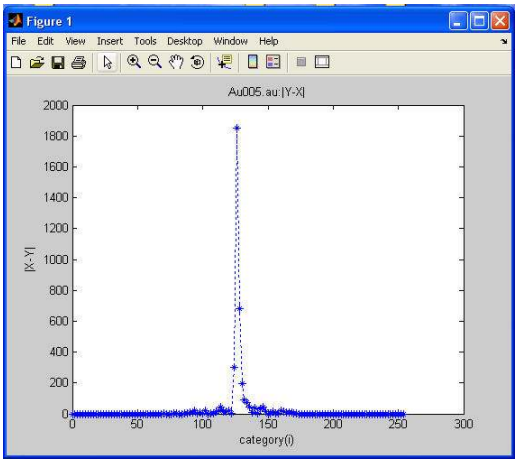


(Au00450)Stego Sese Ait Audio Unitin Spektrumu (%50veri gizlenmiş)

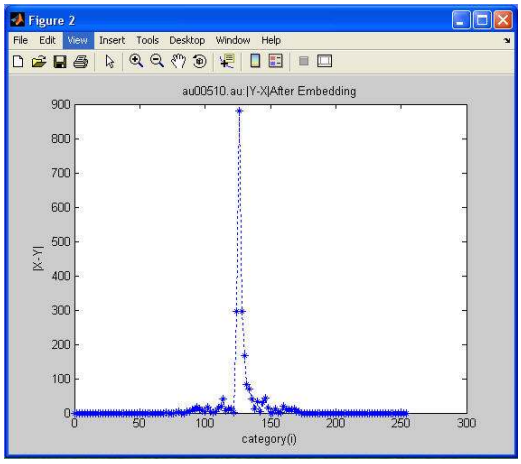


(Au004100)Stego Sese Ait Audio Unitin Spektrumu (%100veri gizlenmiş)

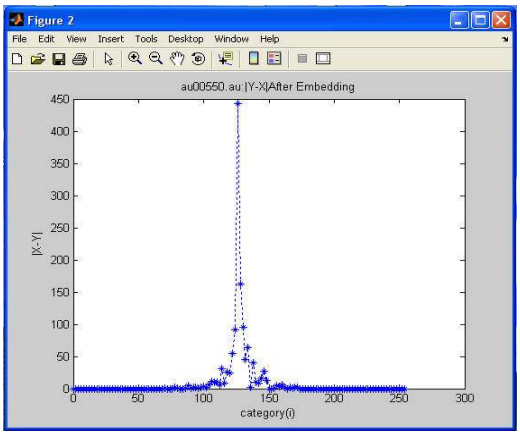
EK-2 (Devam)Ses Dosyaların Görüntüsü



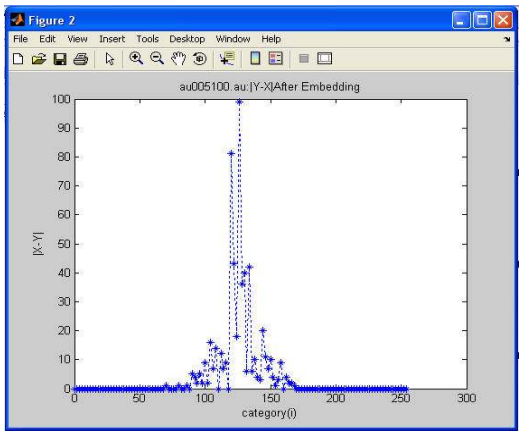
(Au005)Orijinal Sese Ait Audio Unit Histegramı (%0 veri gizlenmiş)



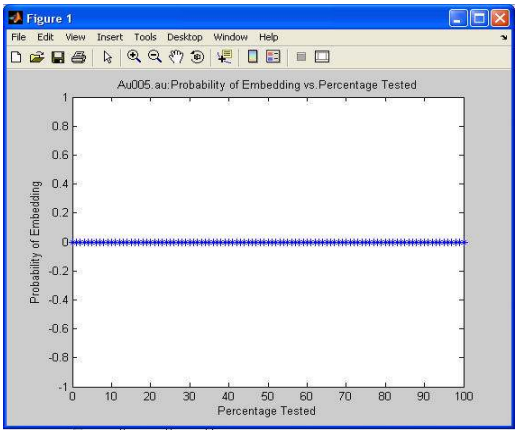
(Au00510)Stego Sese Ait Audio Unit Histegramı (%10 veri gizlenmiş)



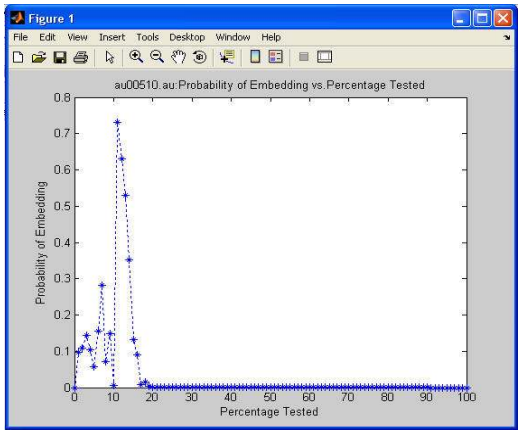
(Au00550)Stego Sese Ait Audio Unit Histegramı (%50 veri gizlenmiş)



(Au005100)Stego Sese Ait Audio Unit Histegramı (%100 veri gizlenmiş)

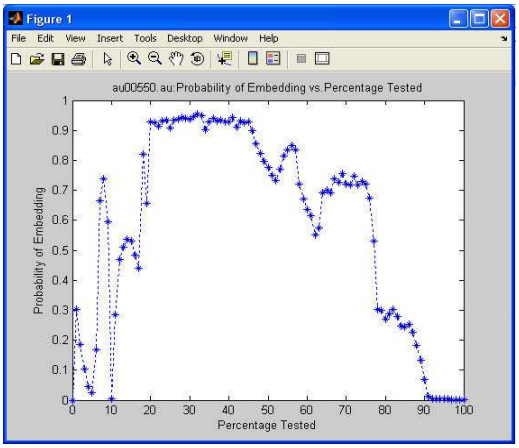


(Au005)Orijinal Sese Ait Audio Unitin Analizi (%0 veri gizlenmiş)

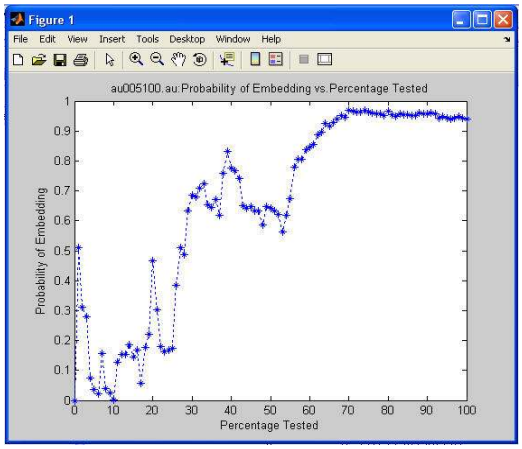


(Au00510)Stego Sese Ait Audio Unitin Analizi (%10 veri gizlenmiş)

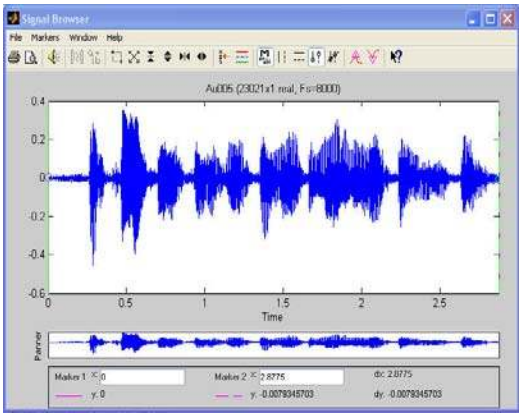
EK-2 (Devam)Ses Dosyaların Görüntüsü



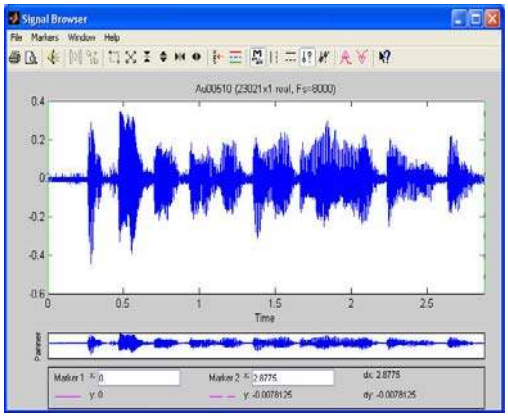
(Au00550)Stego Sese Ait Audio Unitin Analizi (%50 veri gizlenmiş)



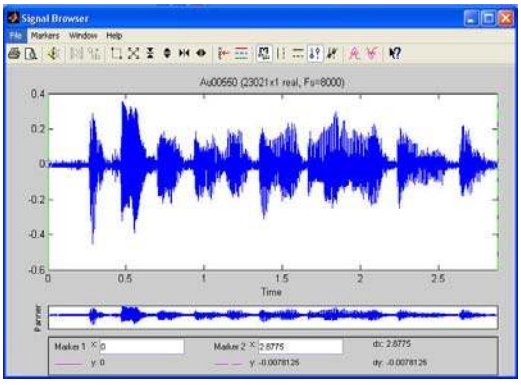
(Au005100)Stego Sese Ait Audio Unitin Analizi (%100 veri gizlenmiş)



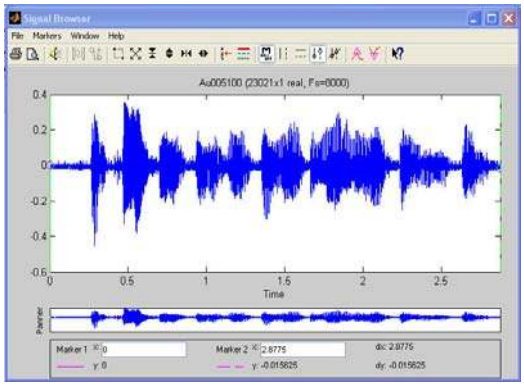
(Au005)Orijinal Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%0 veri gizlenmiş)



(Au00510)Stego Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%10 veri gizlenmiş)

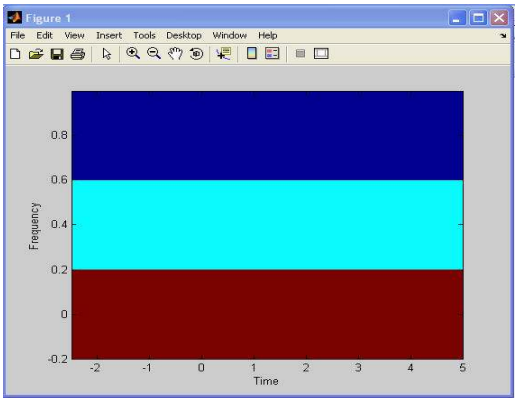


(Au00550)Stego Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%50 veri gizlenmiş)

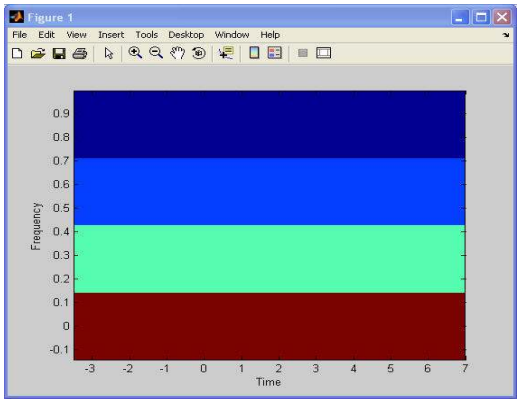


(Au00100)Stego Sese Ait Audio Unitin İşaret Örün Tarayıcısı (%100 veri gizlenmiş)

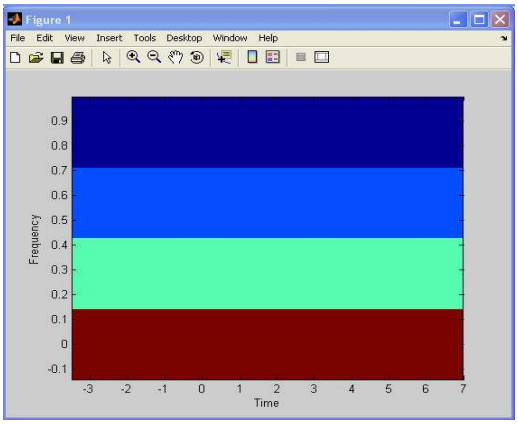
EK-2 (Devam)Ses Dosyaların Görüntüsü



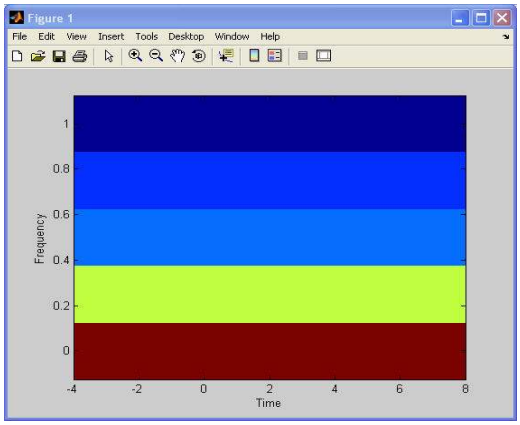
(Au005)Orijinal Sese Ait Audio Unitin Spektrumu (%0 veri gizlenmiş)



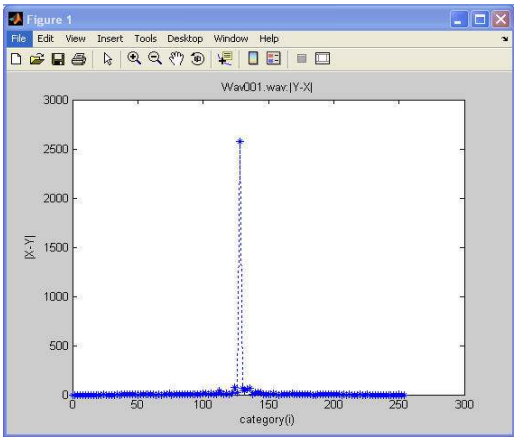
(Au00510)Stego Sese Ait Audio Unitin Spektrumu (%10 veri gizlenmiş)



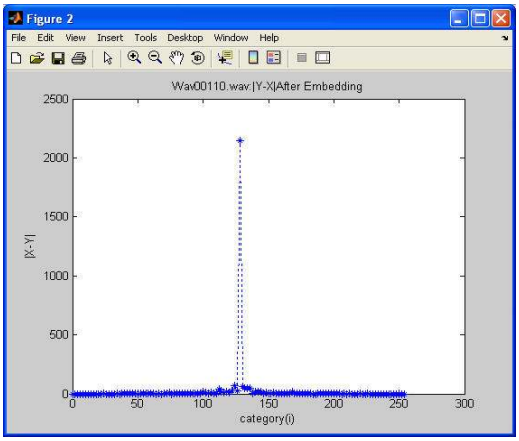
(Au00550)Stego Sese Ait Audio Unitin Spektrumu (%50 veri gizlenmiş)



(Au005100)Stego Sese Ait Audio Unitin Spektrumu (%100 veri gizlenmiş)

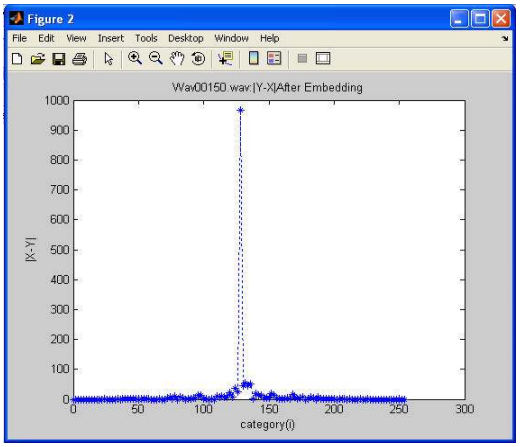


(Wav001) Orijinal Sese Ait Wav Histegramı (%0 veri gizlenmiş)

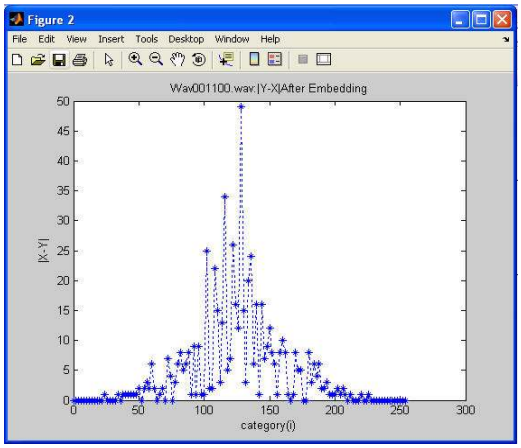


(Wav00110)Stego Sese Ait Wav Histegramı (%10 veri gizlenmiş)

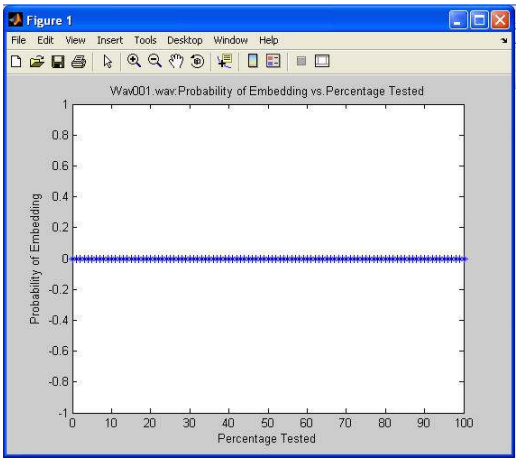
EK-2 (Devam)Ses Dosyaların Görüntüsü



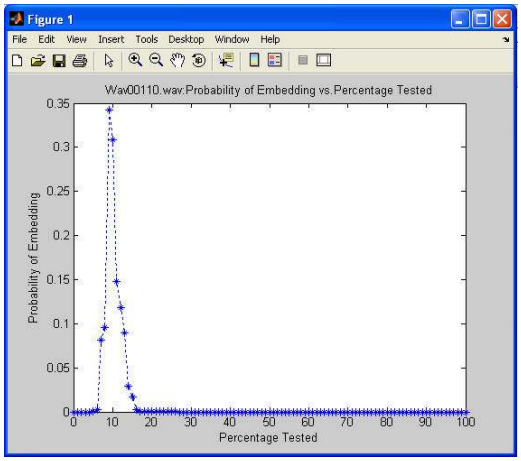
(Wav00150)Stego Sese Ait Wav Histegramı (%10 veri gizlenmiş)



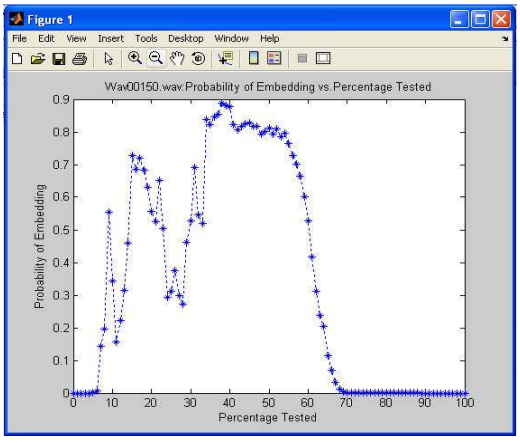
(Wav001100)Stego Sese Ait Wav Histegramı (%100 veri gizlenmiş)



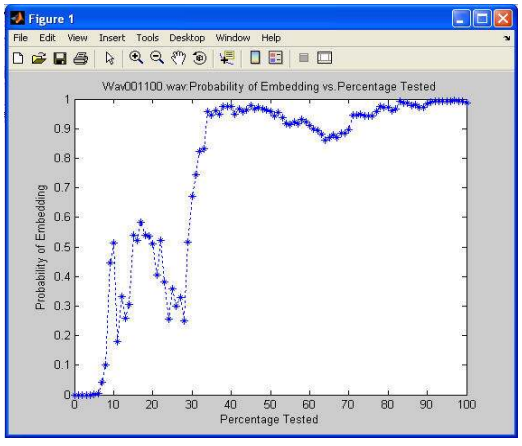
(Wav001)Orijinal Sese Ait Wav Analizi (%0 veri gizlenmiş)



(Wav00110)Stego Sese Ait Wav Analizi (%10 veri gizlenmiş)

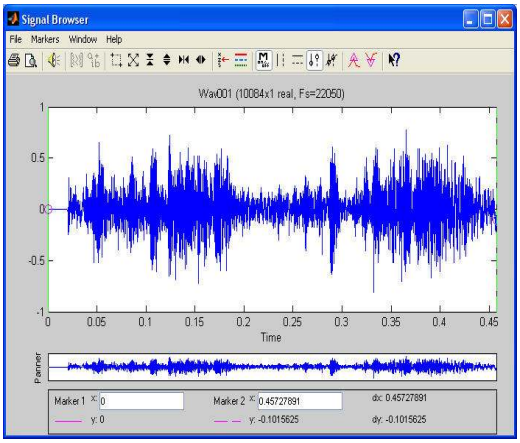


(Wav00150)Stego Sese Ait Wav Analizi (%50 veri gizlenmiş)

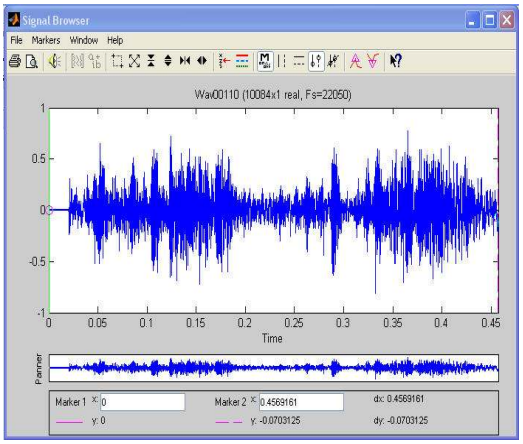


(Wav001100)Stego Sese Ait Wav Analizi (%100 veri gizlenmiş)

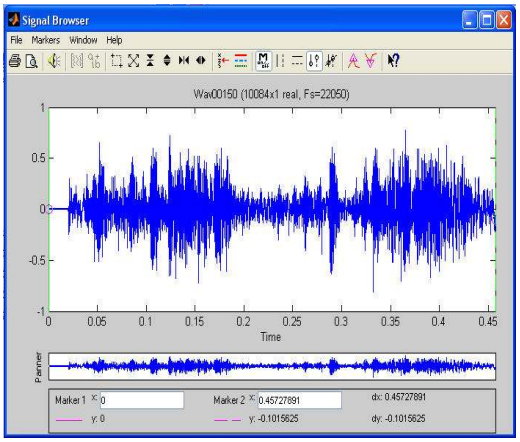
EK-2 (Devam)Ses Dosyaların Görüntüsü



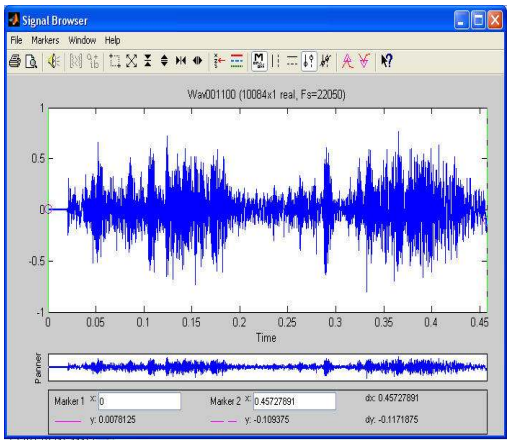
(Wav001)Orijinal Sese Ait Wav İşaret Örün Tarayıcısı (%0 veri gizlenmiş)



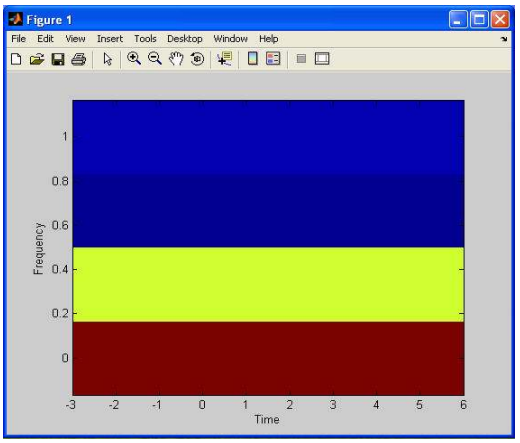
(Wav00110)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%10 veri gizlenmiş)



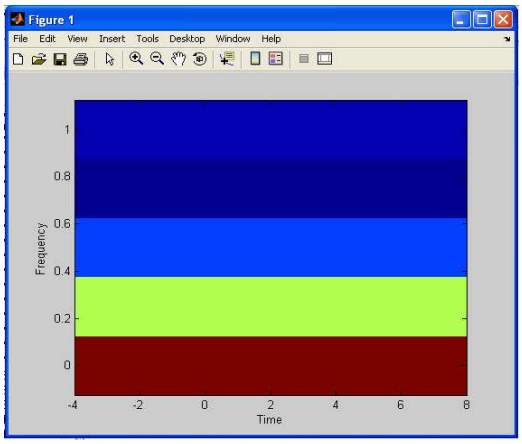
(Wav00150)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%50 veri gizlenmiş)



(Wav001100)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%100 veri gizlenmiş)

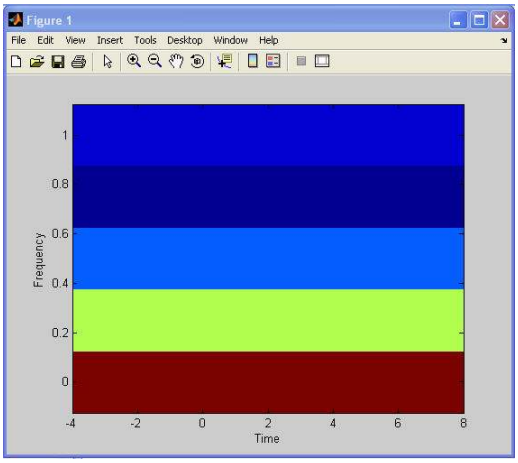


(Wav001)Orijinal Sese Ait Wav Spektrumu (%0 veri gizlenmiş)

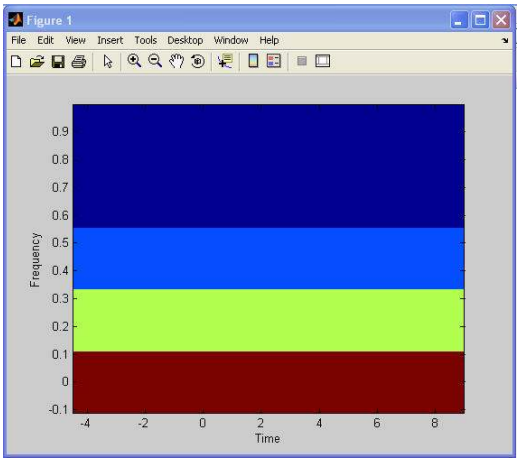


(Wav00110)Stego Sese Ait Wav Spektrumu (%10 veri gizlenmiş)

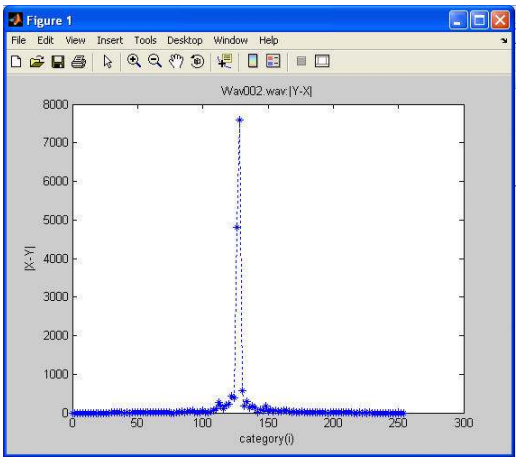
EK-2 (Devam)Ses Dosyaların Görüntüsü



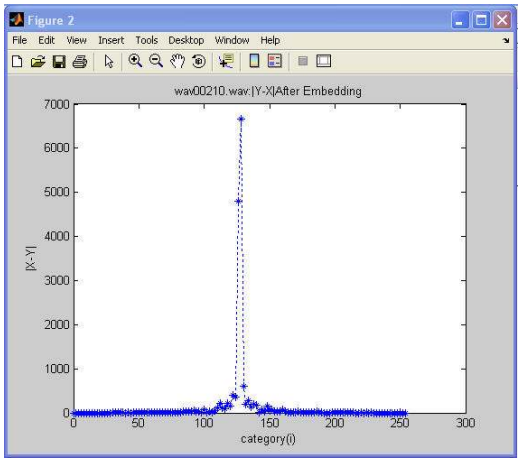
(Wav00150)Stego Sese Ait Wav Spektrumu (%50 veri gizlenmiş)



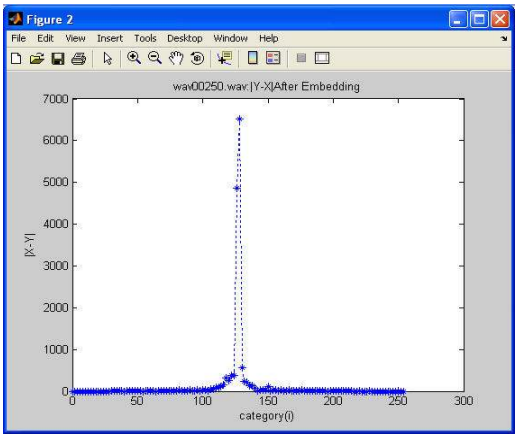
(Wav001100)Stego Sese Ait Wav Spektrumu (%100 veri gizlenmiş)



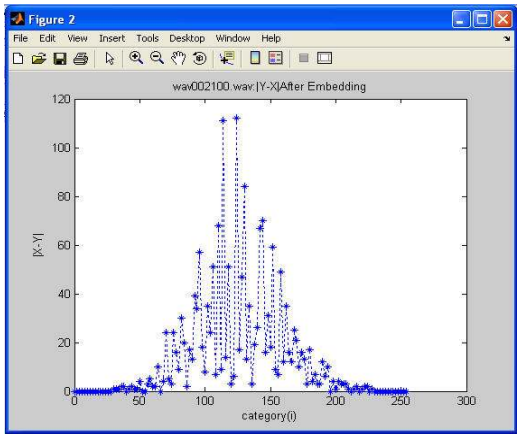
(Wav002)Orijinal Sese Ait Wav Histegramı (%0 veri gizlenmiş)



(Wav00210)Stego Sese Ait Wav Histegramı (%10 veri gizlenmiş)

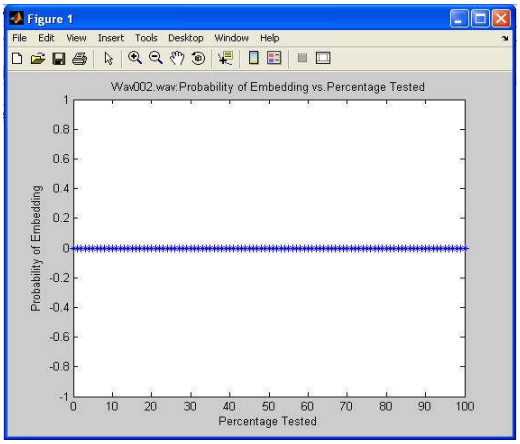


(Wav00250)Stego Sese Ait Wav Histegramı (%50 veri gizlenmiş)

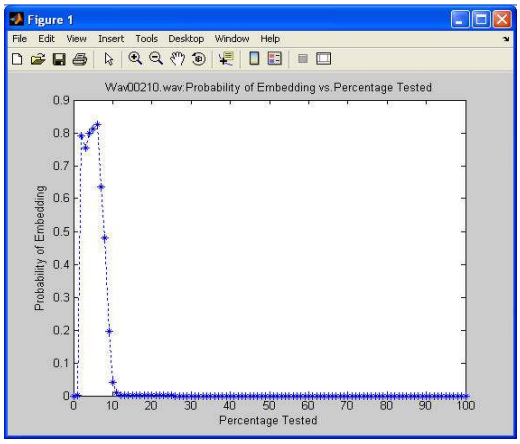


(Wav002100)Stego Sese Ait Wav Histegramı (%100 veri gizlenmiş)

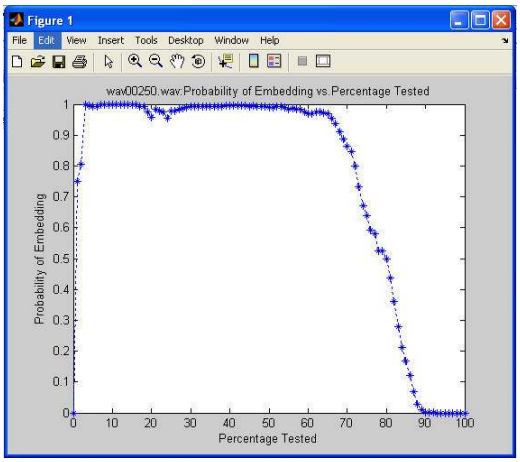
EK-2 (Devam)Ses Dosyaların Görüntüsü



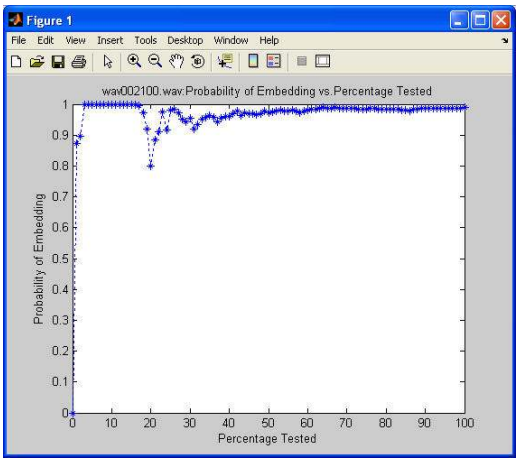
(Wav002)Orijinal Sese Ait Wav Analizi (%0 veri gizlenmiş)



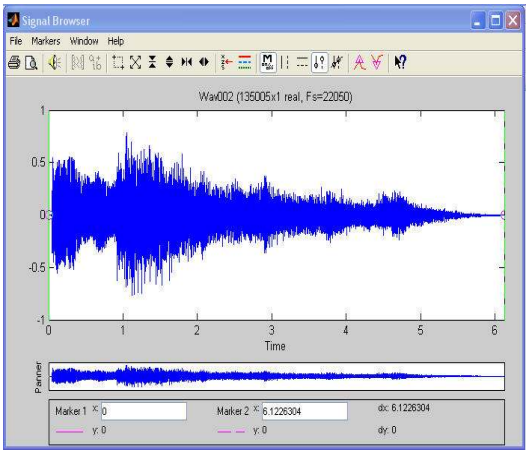
(Wav00210)Stego Sese Ait Wav Analizi (%10 veri gizlenmiş)



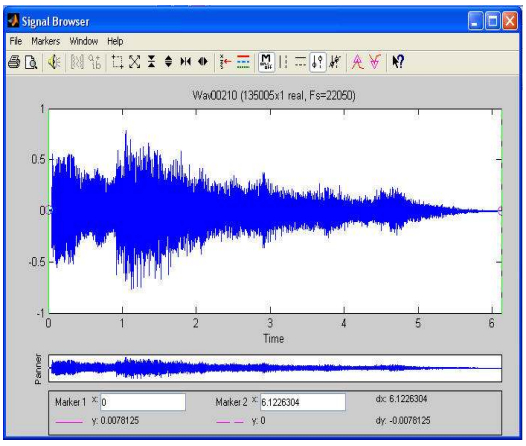
(Wav00250)Stego Sese Ait Wav Analizi (%50 veri gizlenmiş)



(Wav002100)Stego Sese Ait Wav Analizi (%100 veri gizlenmiş)

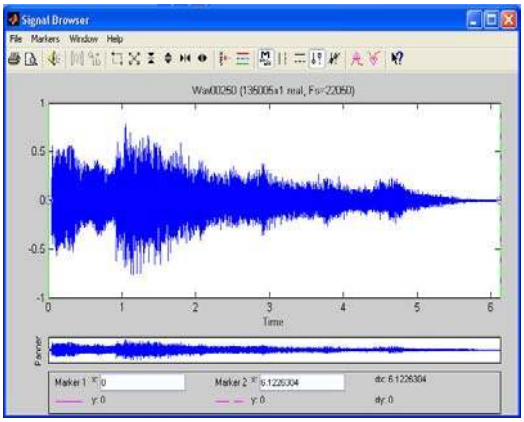


(Wav002)Orijinal Sese Ait Wav İşaret Örün Tarayıcısı (%0 veri gizlenmiş)

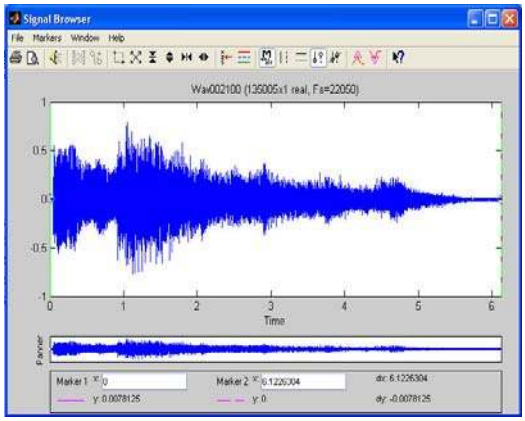


(Wav00210)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%10 veri gizlenmiş)

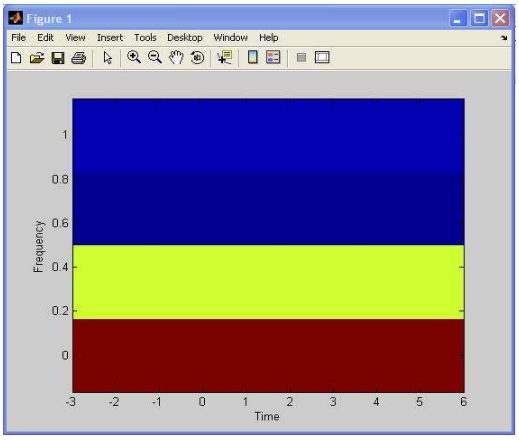
EK-2 (Devam)Ses Dosyaların Görüntüsü



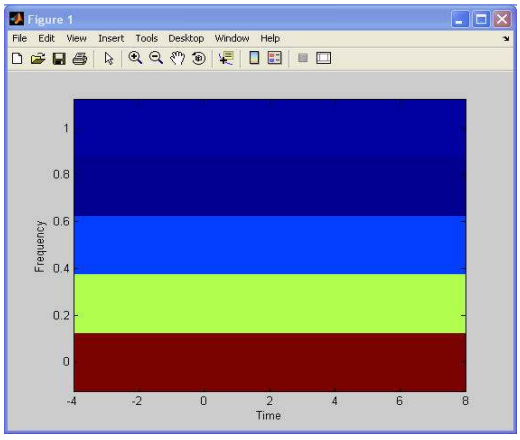
(Wav00250)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%50 veri gizlenmiş)



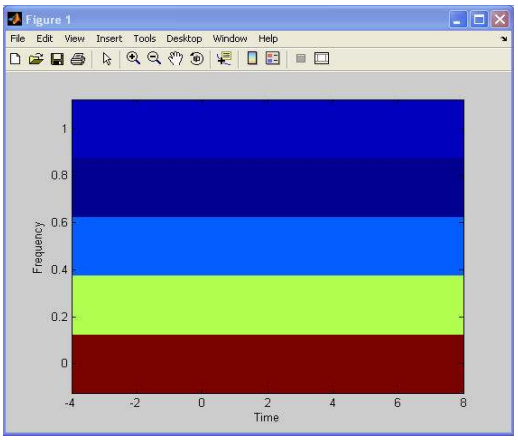
(Wav002100)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%100 veri gizlenmiş)



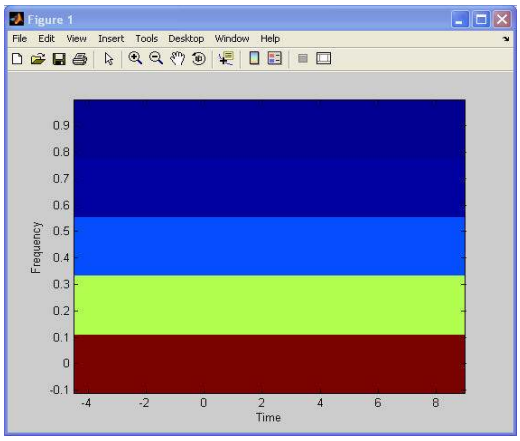
(Wav002)Orijinal Sese Ait Wav Spektrumu (%0 veri gizlenmiş)



(Wav00210)Stego Sese Ait Wav Spektrumu (%10 veri gizlenmiş)

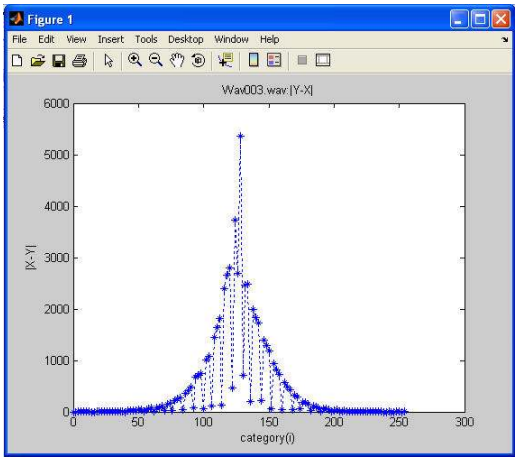


(Wav00250)Stego Sese Ait Wav Spektrumu (%50 veri gizlenmiş)

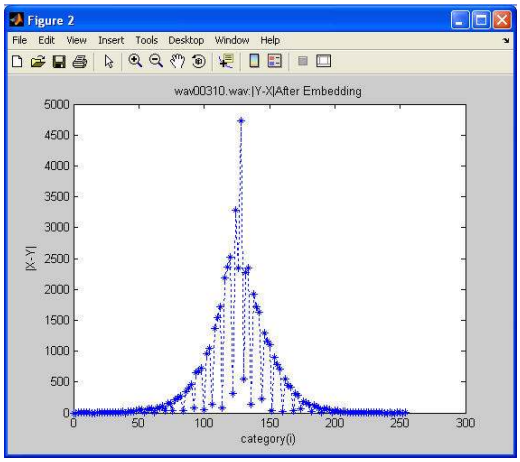


(Wav002100)Stego Sese Ait Wav Spektrumu (%100 veri gizlenmiş)

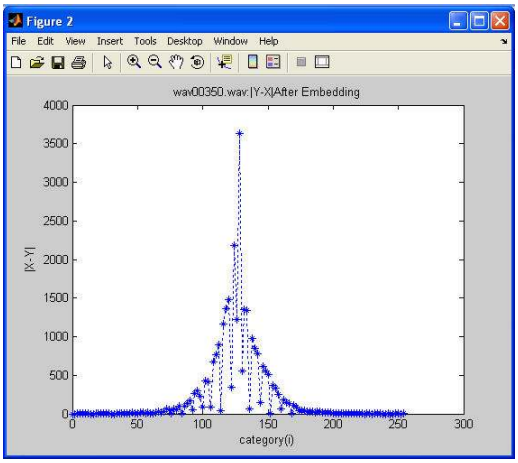
EK-2 (Devam)Ses Dosyaların Görüntüsü



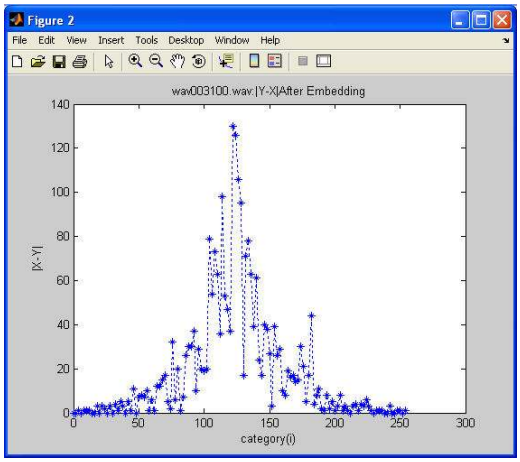
(Wav003)Orijinal Sese Ait Wav Histegramı (%0 veri gizlenmiş)



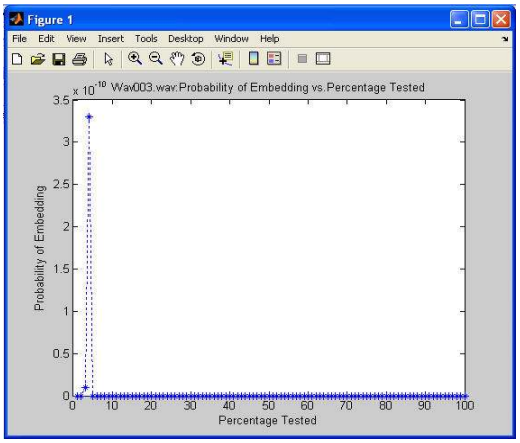
(Wav00310)Stego Sese Ait Wav Histegramı (%10 veri gizlenmiş)



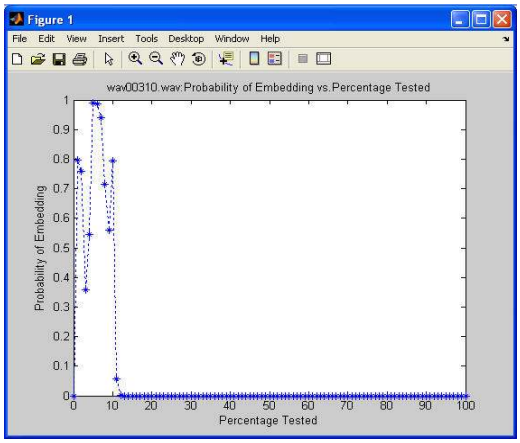
(Wav00350)Stego Sese Ait Wav Histegramı (%50 veri gizlenmiş)



(Wav003100)Stego Sese Ait Wav Histegramı (%100 veri gizlenmiş)

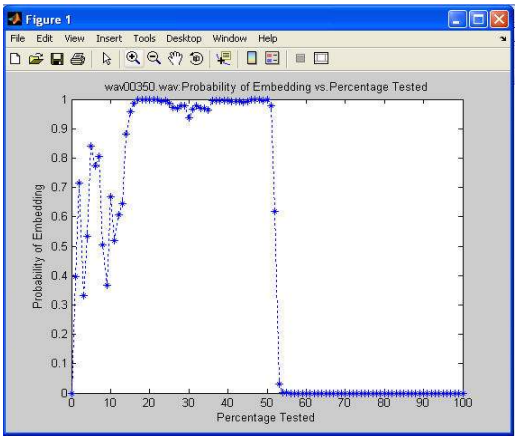


(Wav003)Orijinal Sese Ait Wav Analizi (%0 veri gizlenmiş)

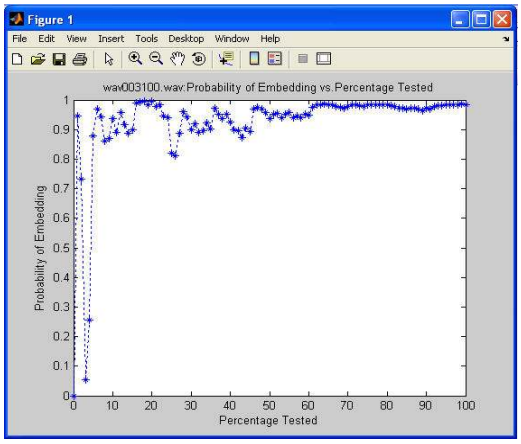


(Wav00310)Stego Sese Ait Wav Analizi (%10 veri gizlenmiş)

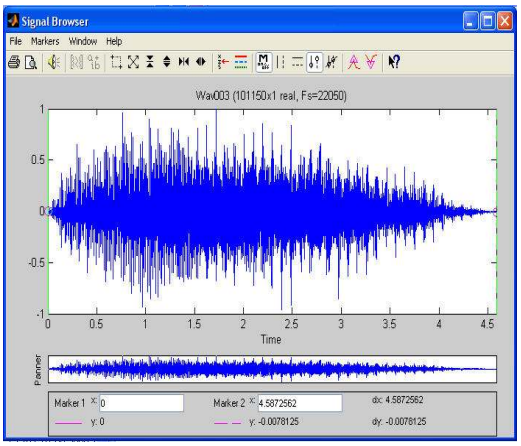
EK-2 (Devam)Ses Dosyaların Görüntüsü



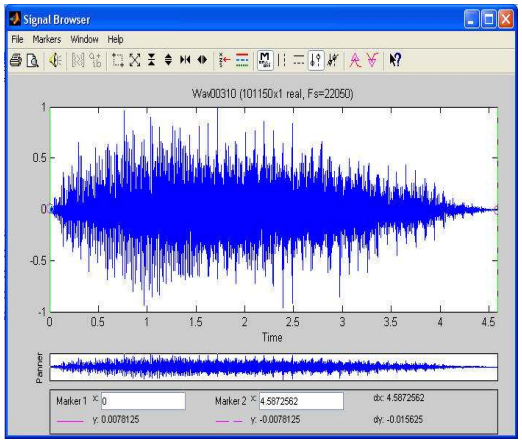
(Wav00350)Stego Sese Ait Wav Analizi (%50 veri gizlenmiş)



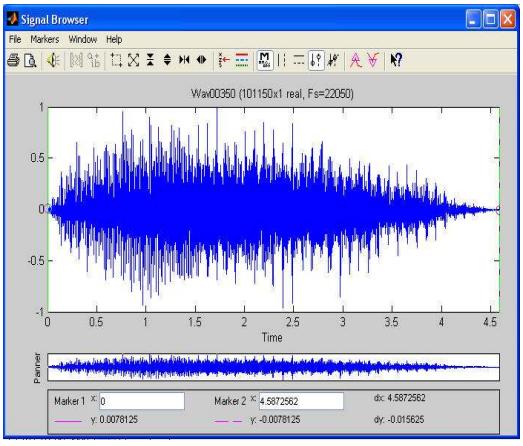
(Wav003100)Stego Sese Ait Wav Analizi (%100 veri gizlenmiş)



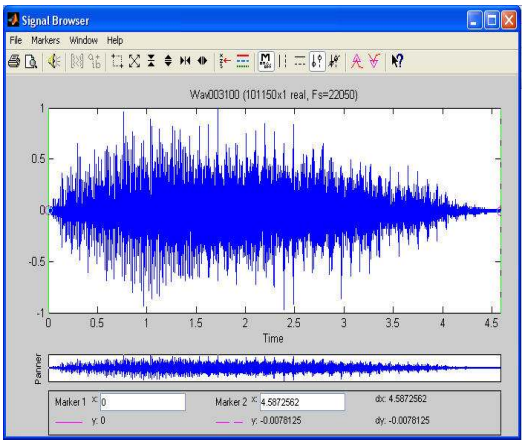
(Wav003)Orijinal Sese Ait Wav İşaret Örün Tarayıcısı (%0 veri gizlenmiş)



(Wav00310)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%10 veri gizlenmiş)

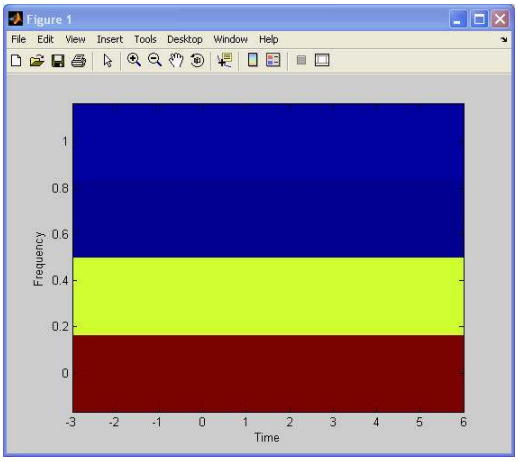


(Wav00350)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%50 veri gizlenmiş)

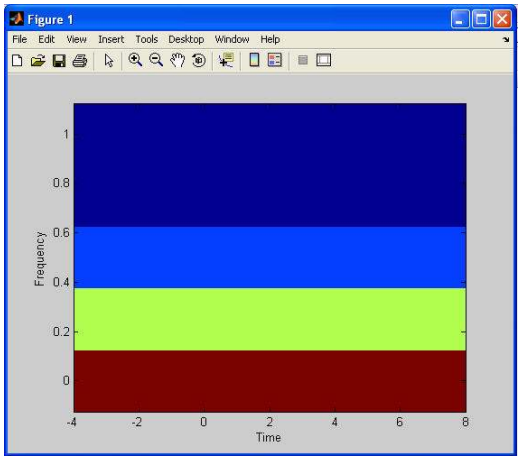


(Wav003100)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%100 veri gizlenmiş)

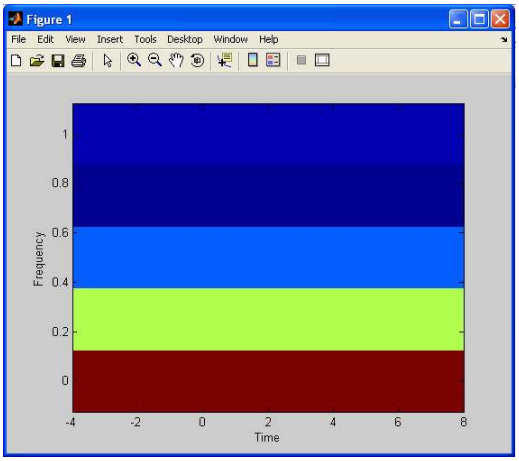
EK-2 (Devam)Ses Dosyaların Görüntüsü



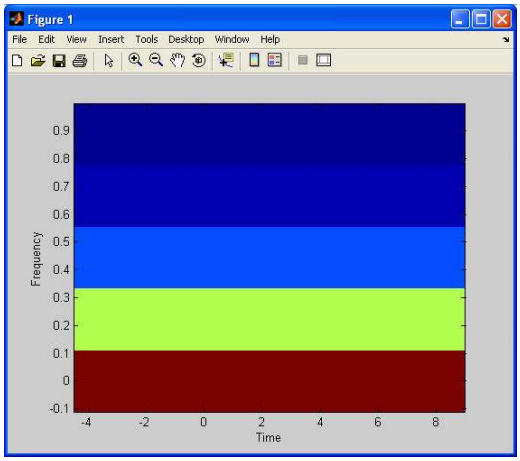
(Wav003)Orijinal Sese Ait Wav Spektrumu (%0 veri gizlenmiş)



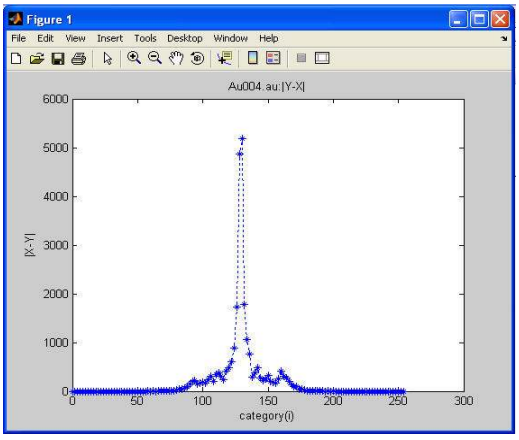
(Wav00310)Stego Sese Ait Wav Spektrumu (%10 veri gizlenmiş)



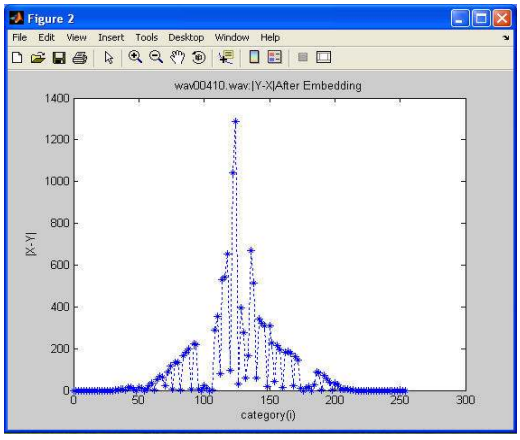
(Wav00350)Stego Sese Ait Wav Spektrumu (%50 veri gizlenmiş)



(Wav003100)Stego Sese Ait Wav Spektrumu (%100 veri gizlenmiş)

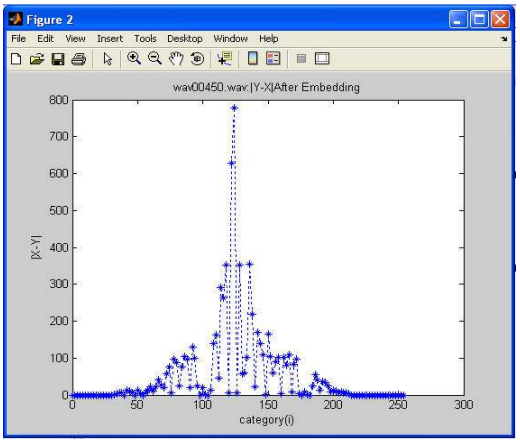


(Au004)Orijinal Sese Ait Audio Unit Histegramı (%0 veri gizlenmiş)

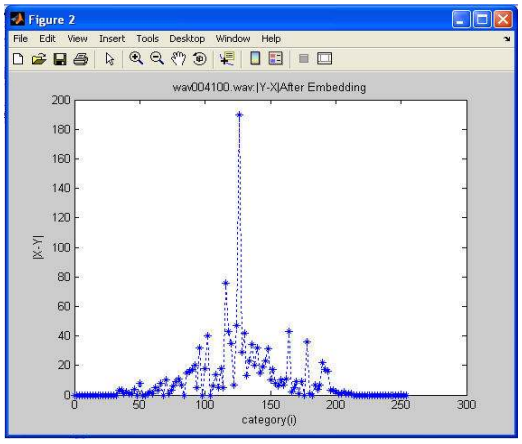


(Wav00410)Stego Sese Ait Wav Histegramı (%10 veri gizlenmiş)

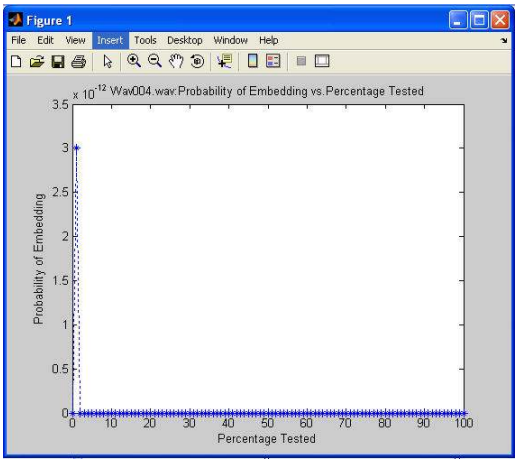
EK-2 (Devam)Ses Dosyaların Görüntüsü



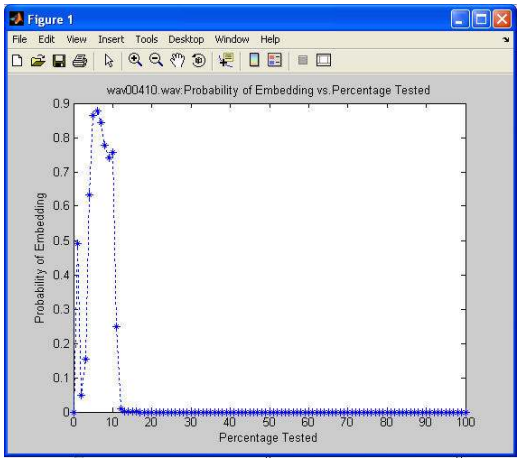
(Wav00450)Stego Sese Ait Wav Histegramı (%50 veri gizlenmiş)



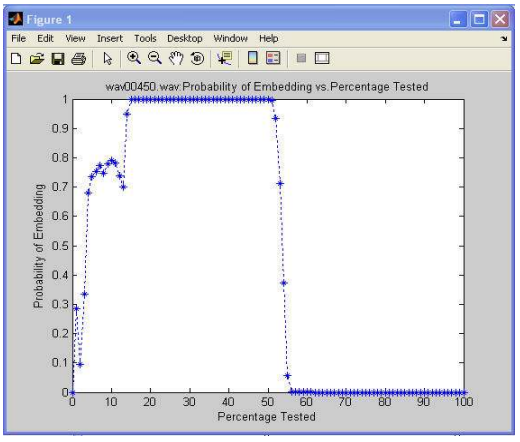
(Wav004100)Stego Sese Ait Wav Histegramı (%100 veri gizlenmiş)



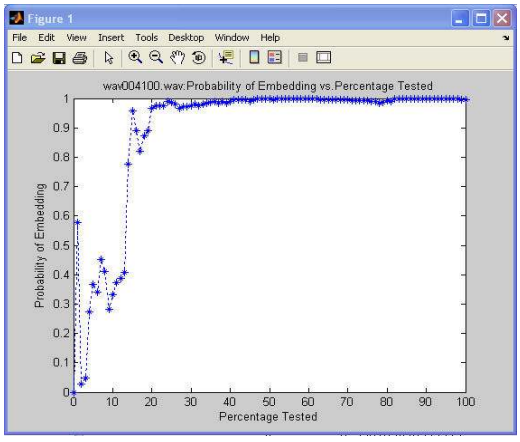
(Wav004)Orijinal Sese Ait Wav Analizi (%0 veri gizlenmiş)



(Wav00410)Stego Sese Ait Wav Analizi (%10 veri gizlenmiş)

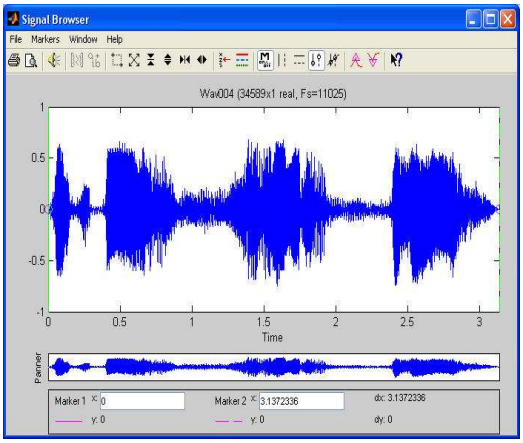


(Wav00450)Stego Sese Ait Wav Analizi (%50 veri gizlenmiş)

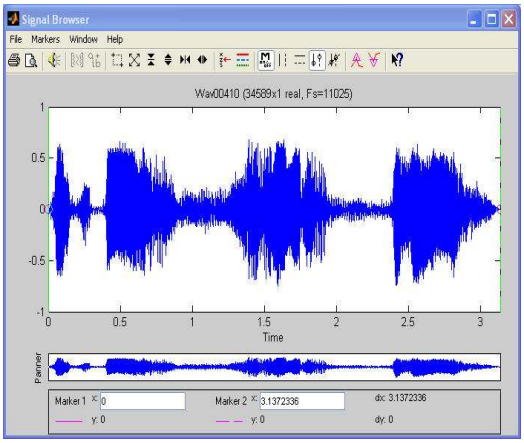


(Wav004100)Stego Sese Ait Wav Analizi (%100 veri gizlenmiş)

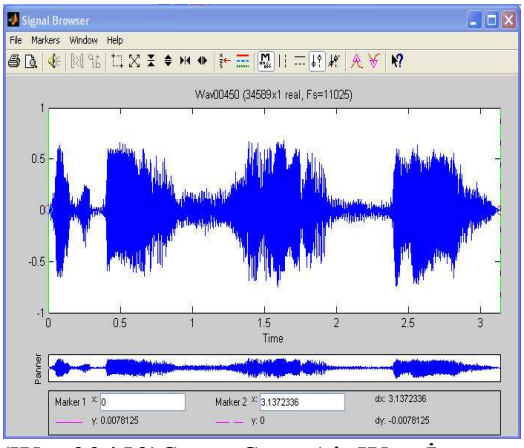
EK-2 (Devam)Ses Dosyaların Görüntüsü



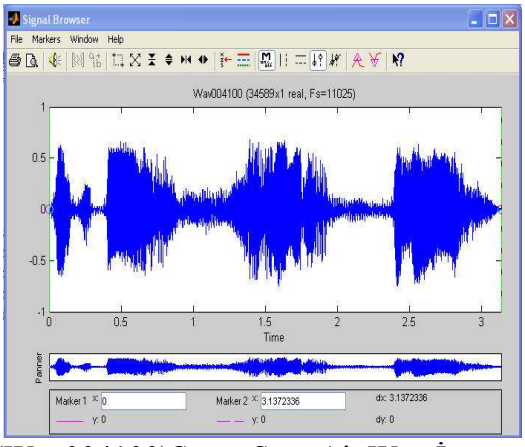
(Wav004)Orijinal Sese Ait Wav İşaret Örün Tarayıcısı (%0 veri gizlenmiş)



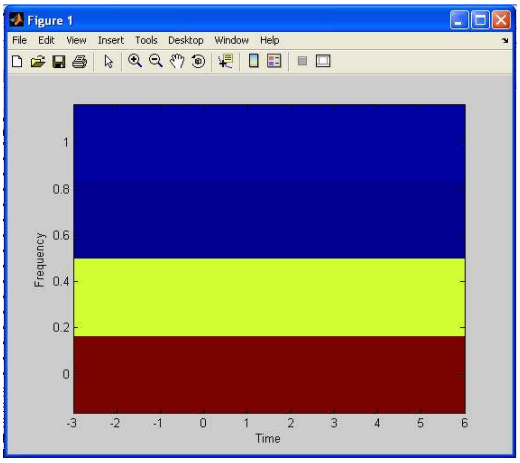
(Wav00410)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%10 veri gizlenmiş)



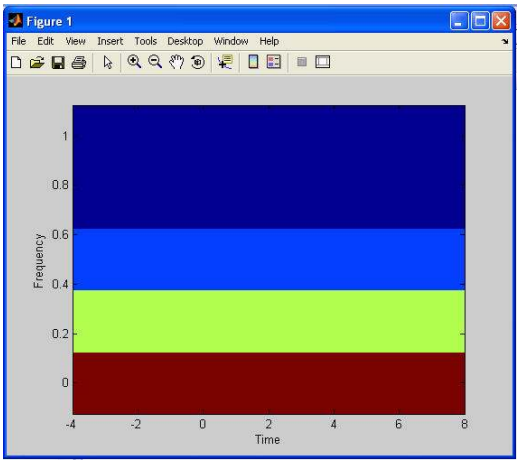
(Wav00450)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%50 veri gizlenmiş)



(Wav004100)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%100 veri gizlenmiş)

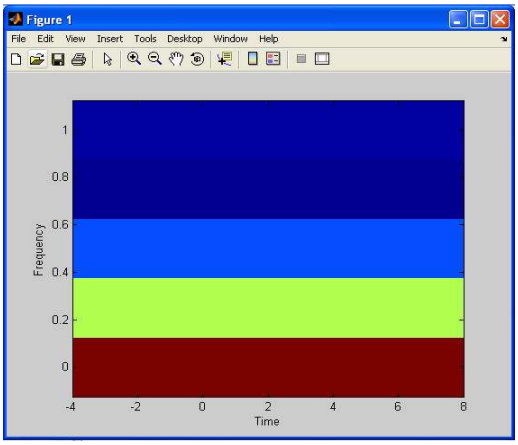


(Wav004)Orijinal Sese Ait Wav Spektrumu (%0 veri gizlenmiş)

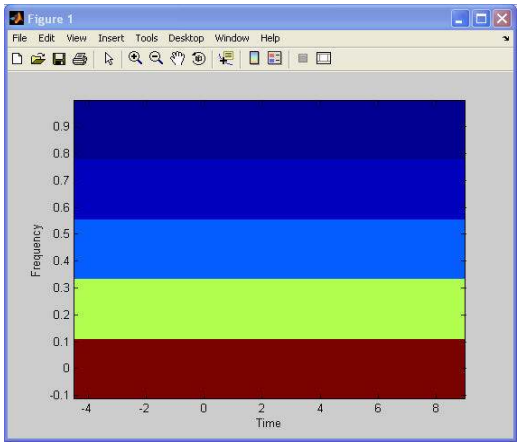


(Wav00410)Stego Sese Ait Wav Spektrumu (%10 veri gizlenmiş)

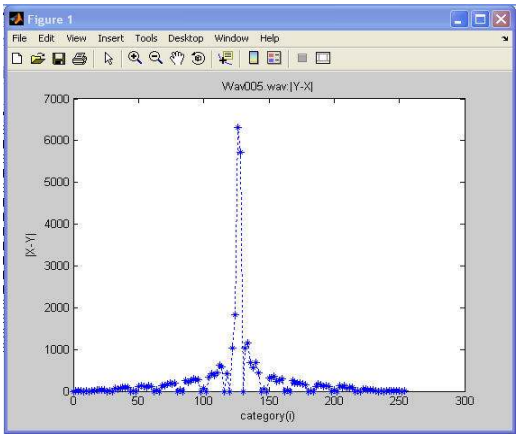
EK-2 (Devam)Ses Dosyaların Görüntüsü



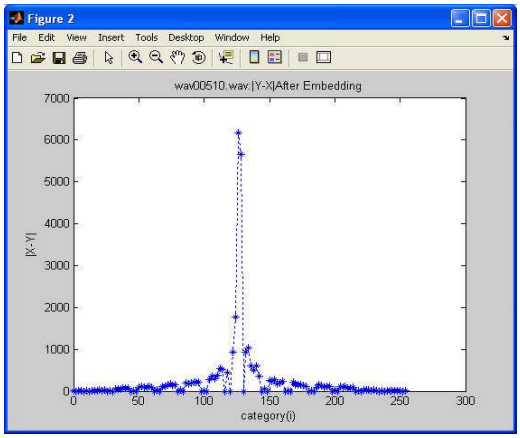
(Wav00450)Stego Sese Ait Wav Spektrumu (%50 veri gizlenmiş)



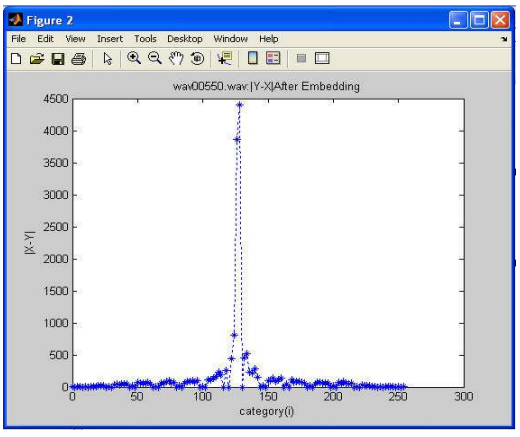
(Wav004100)Stego Sese Ait Wav Spektrumu (%100 veri gizlenmiş)



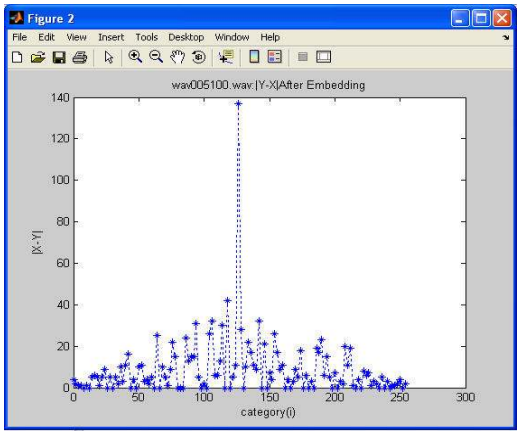
(Wav005)Orijinal Sese Ait Wav Histegramı (%0 veri gizlenmiş)



(Wav00510)Stego Sese Ait Wav Histegramı (%10 veri gizlenmiş)

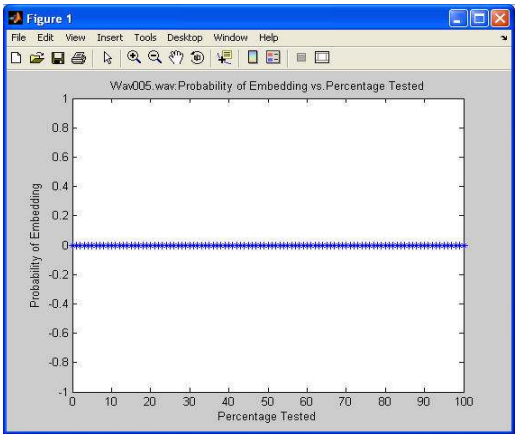


(Wav00550)Stego Sese Ait Wav Histegramı (%50 veri gizlenmiş)

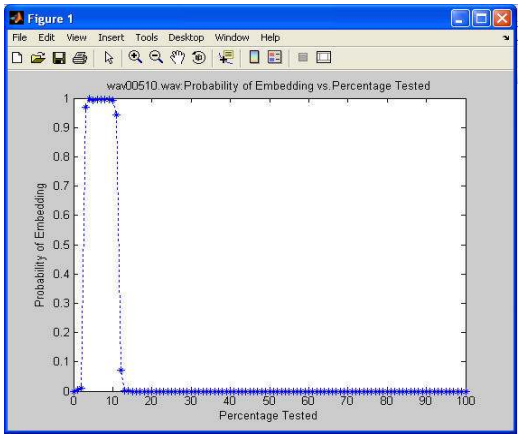


(Wav005100)Stego Sese Ait Wav Histegramı (%100 veri gizlenmiş)

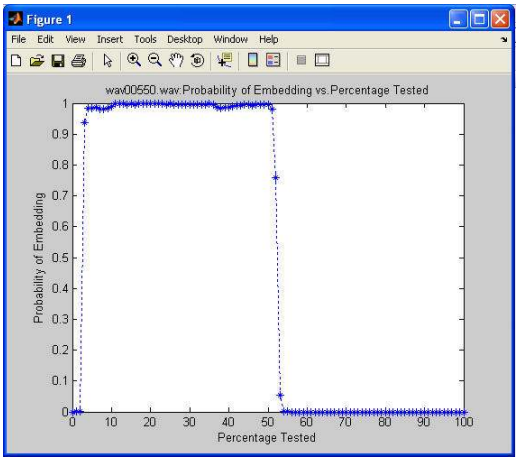
EK-2 (Devam)Ses Dosyaların Görüntüsü



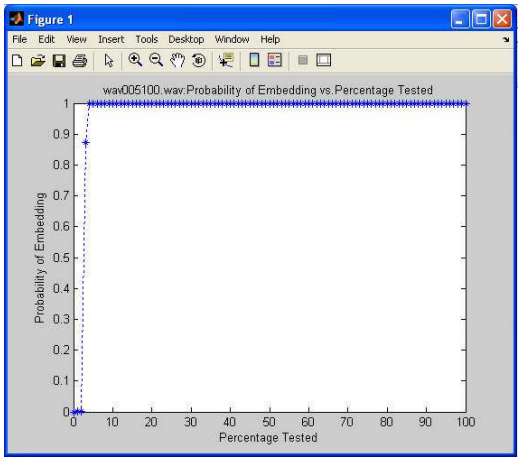
(Wav005)Orijinal Sese Ait Wav Analizi (%0 veri gizlenmiş)



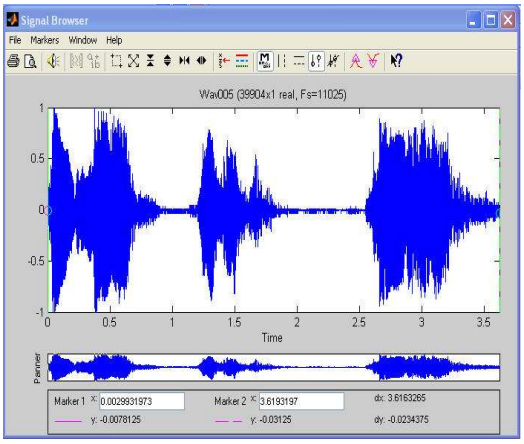
(Wav00510)Stego Sese Ait Wav Analizi (%10 veri gizlenmiş)



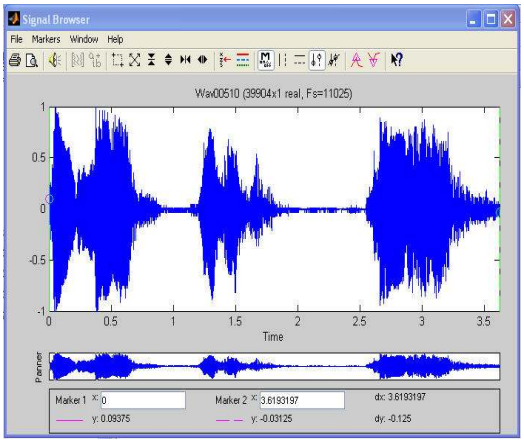
(Wav00550)Stego Sese Ait Wav Analizi (%50 veri gizlenmiş)



(Wav005100)Stego Sese Ait Wav Analizi (%100 veri gizlenmiş)

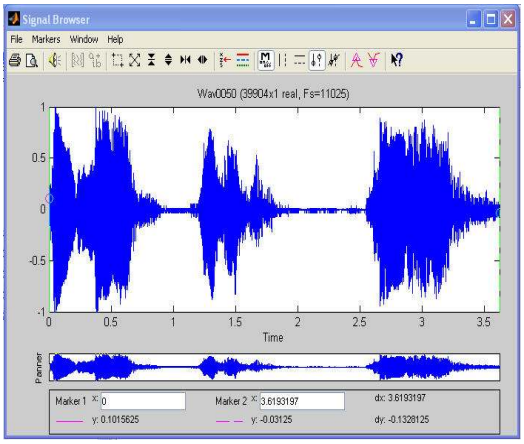


(Wav005)Orijinal Sese Ait Wav İşaret Örün Tarayıcısı (%0 veri gizlenmiş)

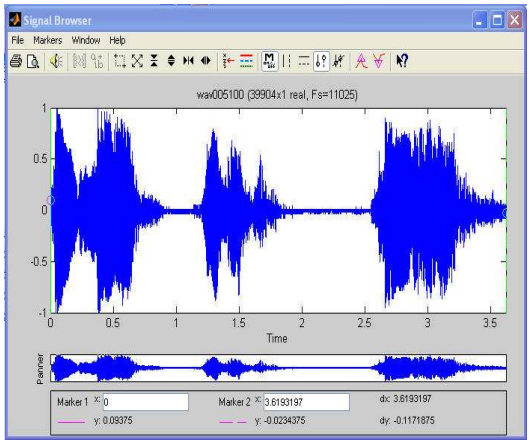


(Wav00510)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%10 veri gizlenmiş)

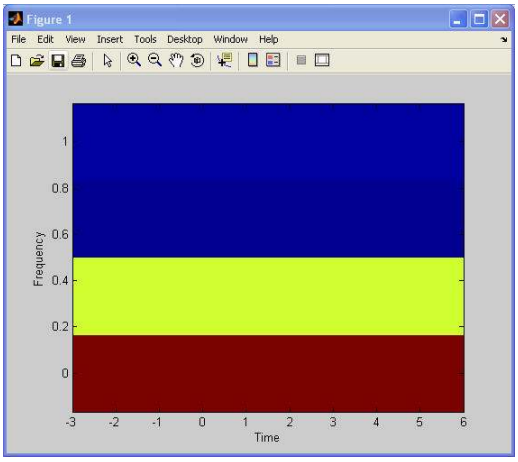
EK-2 (Devam)Ses Dosyaların Görüntüsü



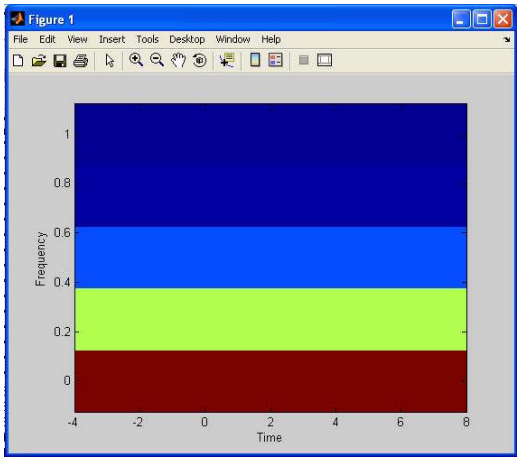
(Wav00550)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%50 veri gizlenmiş)



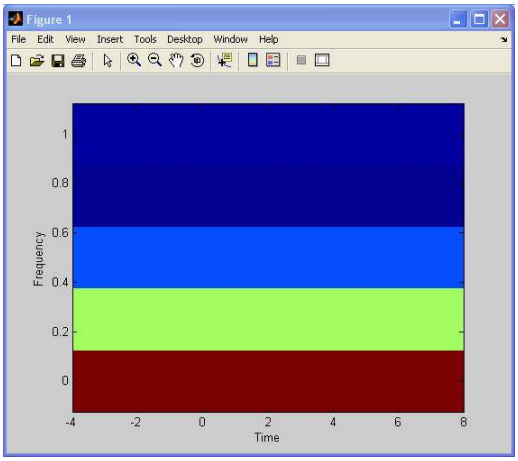
(Wav005100)Stego Sese Ait Wav İşaret Örün Tarayıcısı (%100 veri gizlenmiş)



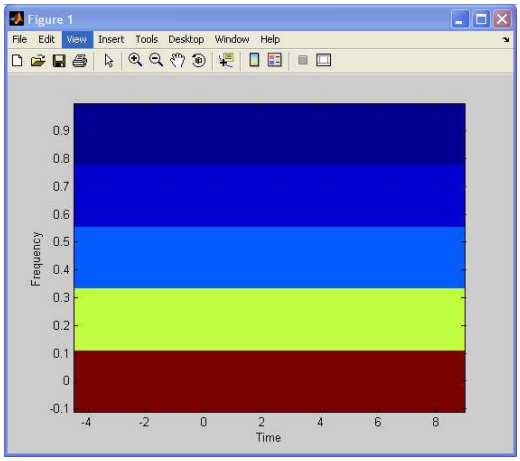
(Wav005)Orijinal Sese Ait Wav Spektrumu (%0 veri gizlenmiş)



(Wav00510)Stego Sese Ait Wav Spektrumu (%10 veri gizlenmiş)



(Wav00550)Stego Sese Ait Wav Spektrumu (%50 veri gizlenmiş)



(Wav005100)Stego Sese Ait Wav Spektrumu (%100 veri gizlenmiş)

EK-3 Matlab Code

3.1 FE aracı.m

```
%FE aracı

%Gömme algorithm

function stego=flipembed
image_name = input('Enter name and location of COVER IMAGE (exclude extension): ', 's');
%Location and name of image in which to hide data
image_type = input('Enter image type, e.g., png,bmp: ', 's');
%Input image type input_image = imread(image_name, image_type);
%Read input image
output_image = input('Enter name and location of
%output image (exclude extension): ', 's');
answer = input('Histograms? y or n: ', 's');
original = double(input_image);
cover = double(input_image);
%Converts cover image to double precision matrix for computations
[p,q] = size(cover);
Xa = zeros(128,1);
Xb = zeros(128,1);
Ya = zeros(128,1);
Yb = zeros(128,1);
%---------------------------------------------------
%---------------------------------------------------
%Histogram

for i=1:p
   for j=1:q
            if rem(cover(i,j),2)==0
              Xb((cover(i,j)/2)+1)=Xb((cover(i,j)/2)+1)+1;
            else
              Yb(((cover(i,j)-1)/2)+1)=Yb(((cover(i,j)- 1)/2)+1)+1;
            end
          end
end
if answer == 'y'
  categories = zeros(128,1);
  for i = 1:128
     categories(i) = 2*(i-1);
 end
 V = abs(Yb-Xb);
fig_title = [image_name, '.', image_type,':','|Y-X|  Before Embedding'];
 figure (1), plot(categories, V, '*:');  title(fig_title);
 xlabel('Category (i)'); ylabel('|Y-X|');

%displays graph
end
%---------------------------------------------------
%---------------------------------------------------
%random message generator
length = input('Enter length of desired message (in percent--use integers 1-100 only): ');
while ((length < 1) | (100 < length))
```

EK-3 (Devam) Matlab Code

```
if (100 < length)
length = input('Message too large for cover image. Enter length of desired message( in percent—use integers 1-100 only): ');
else
length = input('Invalid message length. Enter length of desired message( in percent—use integers 1-100 only): ');
end
end
length = floor((length/100)*p*q);
save('length','length');
M1 = rand(p,q);
M1 = rem(floor(10*M1),2);
rows = floor(length/q);
col = length - (rows*q);
stego = floor(cover/2)*2 + M1;
if rows < p
nextrow = cat(2,stego(rows+1,1:col),cover(rows+1, col+1:q));
end
switch rows
case p
stego = floor(cover/2)*2 + M1;
case (p-1)
stego = cat(1, stego(1:rows,1:q),nextrow);
otherwise
stego = cat(1, stego(1:rows,1:q),nextrow,cover(rows+2:p,1:q));
end
save('M1','M1');
%-------------------------------------------------------
%-------------------------------------------------------
%Histogram
for i=1:p
for j=1:q
if rem(stego(i,j),2)==0
Xa((stego(i,j)/2)+1)=Xa((stego(i,j)/2)+1)+1;
else
Ya(((stego(i,j)-1)/2)+1)=Ya(((stego(i,j)-1)/2)+1)+1;
end
end
end
if answer == 'y'
W = abs(Ya-Xa);
fig_title = [output_image, '.', image_type,': ','|Y-X| After Embedding'];
figure (2), plot(categories,W,'*:'); title(fig_title);
xlabel('Category (i)'); ylabel('|Y-X|');
%displays graph
end
%-------------------------------------------------------
%display(stego);
save('stego','stego');
stego1 = uint8(stego);
save('stego1','stego1');
save('original','original');
imwrite(stego1,output_image,'png');
```

EK-3 (Devam) Matlab Code

3.2 Gömmek2.m

When the pixel values in the cover image do not have the same parity as the corresponding message bit, Embed2.m decrements the pixel values in the cover image to obtain the pixel values in the stego image. Embed2.m then takes the absolute value of all of the pixels so that if a zero in the cover image is decremented into a -1 in the stego image, the -1 becomes +1.21 The code for Embed2.m follows:

```
%gömmek2 function
%gömme algorithm
function stego=embed2
%(x) again=1;
%while again==1
image_name = input('Enter name and location of COVER IMAGE (exclude extension): ', 's');
%Location and name of image in which to hide data
image_type = input('Enter image type, e.g., png,bmp: ', 's');
%Input image type
input_image = imread(image_name, image_type);
%Read input image
output_image = input('Enter name and location of output image (exclude extension): ', 's');
original = double(input_image);
cover = double(input_image);
%Converts cover image In order to satisfy the image format requirements, all pixel %values must be positive. to double precision matrix for computations message = %double(x); Converts message to double precision matrix for computations
[p,q] = size(cover);
%-----------------------------------------------------
%-----------------------------------------------------
%random message generator
length = input('Enter length of desired message (in percent--use integers 1-100 only): ');
length = floor((length/100)*p*q);
save('length','length');
%display(length);
%return;
while ((p*q) < length)
length = input('Message too large for cover image. Enter length of desired message (in bits): ');
end
M1 = rand(p,q);
%pxq matrix with random entries
M1 = rem(floor(10*M1),2);
%Converts matrix of real numbers ranging from 0 to 1 into a matrix of integers from %0 to 9, then converts random numbers from M1 to random bit values--0 if random %number is even, 1 if random number is odd
rows = floor(length/q);
col = length - (rows*q);
if rows < p
nextrow = cat(2,M1(rows+1,1:col),cover(rows+1,col+1:q));
end
switch rows
case p
message = M1
case (p-1)
message = cat(1, M1(1:rows,1:q),nextrow)
otherwise
```

EK-3 (Devam) Matlab Code

```
message = cat(1, M1(1:rows,1:q),nextrow,cover(rows+2:p,1:q))
end
Mess = message;
save('M1','M1');
save('Mess','Mess');
%-------------------------------------------------------
%-------------------------------------------------------
stego = cover - rem((message + cover),2);
stego = abs(stego);
%display(stego);
save('stego','stego');
stego1 = uint8(stego);
save('stego1','stego1');
save('original','original');
imwrite(stego1,output_image,'png');
```

EK-3 (Devam) Matlab Code

3.3 DÇ2.m

```
%DÇ2
image_name = input('Enter image name and location (exclude extension): ', 's'); %Location and name of image to be analyzed
image_type = input('Enter image type, e.g., png,bmp: ', 's'); %Input image type
percent = input('Enter percentage of pixels to test--integers 1-100 only: ');
input_image = imread(image_name, image_type);
%Read input image
A = double(input_image);
%Converts input_image to double precision matrix for computations
k = 128;
[m,n] = size(A); %Allows images of varying sizes
%and will allow FOR loop to pass through all pixels in input image.
%--------------------------------------------------
%--------------------------------------------------
%Percentage function: Asks user for percentage of pixels to test and performs %necessary calculations for 'for' loops to follow. Three while loops require percent %variable to be an integer between 1 and 100, inclusive.
while ((isa(percent, 'int8')==0) & ((percent < 1)| (percent > 100)))
%Requires input to be an integer
percent = input('Invalid entry. Enter percentage of pixels to test--integers 1-100 only: ');
end
percent = double(percent)/100;
%Converts percent variable to float point so the percentage is in decimal form, i.e., %10%=0.10
total_pixels = floor(percent*m*n);
%Calculates total number of pixels to test
whole_rows = floor(total_pixels/n);
%Calculates # of rows to test necessary number of total_pixels
%(for use in the for loop)
columns_last_row = total_pixels - (whole_rows*n);
%Calculates # of columns (elements) in last (possibly partial) row to test necessary %number of total_pixels (for use in the for loop)
%--------------------------------------------------
%--------------------------------------------------
X=zeros(k,1);
%Initialize 128x1 vector to zero. X will count frequencies of even-valued colors.
Y=zeros(k,1);
%Initialize 128x1 vector to zero. Y will count frequencies of odd-valued colors.
D=zeros(k,1);
%Initialize 128x1 vector to zero. D will assist in calculation of Chi-squared statistic.
%Following loop counts the frequencies of color values in pixels of input image. If %the pixel value is even, i.e., == 2i,then the (i+1)th row of X is incremented. If the %pixel value is odd, i.e.,== 2i+1, then the (i+1)th row of Y is incremented. The %(i+1)th row is used because the vectors X and Y are by default indexed from 1 to %128.
for q=1:whole_rows
for r=1:n
if rem(A(q,r),2)==0
X((A(q,r)/2)+1)=X((A(q,r)/2)+1)+1;
else
Y(((A(q,r)-1)/2)+1)=Y(((A(q,r)-1)/2)+1)+1;
end
end
```

EK-3 (Devam) Matlab Code

```
end
for q=whole_rows+1
for r=1:columns_last_row
if rem(A(q,r),2)==0
X((A(q,r)/2)+1)=X((A(q,r)/2)+1)+1;
else
Y(((A(q,r)-1)/2)+1)=Y(((A(q,r)-1)/2)+1)+1;
end
end
end
for i=1:128
if (X(i)+Y(i)) < 5
X(i) = 0;
Y(i) = 0;
k = k-1;
end
end
Z = (X + Y);
Z=Z./2;
%Vector of the averages of X_i and Y_i:
Z_i = (X_i + Y_i)/2.
D = (X-Z).^2;
for i=1:128
if Z(i)==0
D(i) = 0;
else
D(i) = D(i)/Z(i);
end
end
format long g;
C=sum(D);
%C is the Chi-squared statistic, the sum of the elements in D.
display(C);
%Outputs C to the screen.
display(k);
%Begin debugging code
%display(m);
%display(n);
%display(percent);
%display(total_pixels);
%display(whole_rows);
%display(columns_last_row);
%End debugging code
%P=1 - (quadl(@integrand,0,C,1.e-15));
%Calculates probability that image has an embedded message.
Pcheck=1-gammainc(C/2,(k-1)/2);
%Checks value calculated for P using gammainc function.
%display(P);
%Outputs probability P to the screen.
display(Pcheck);
%Outputs Pcheck to screen.
```

EK-3 (Devam) Matlab Code

3.4 DÇ2r.m

```
%DÇ2r

%DÇ2 reverse

%[image_name, image_type] = textread('images.dat','%s %s');
image_name = input('Enter image name and location (exclude extension): ', 's'); %Location and name of image to be analyzed
image_type = input('Enter image type, e.g., png, bmp:', 's');
%Input image type
begin_percent = input('Enter percentage of pixels to SKIP (at the beginning of the image)--integers1-100 only: ');
end_percent = input('Enter percentage of pixels to test (beginning after the skipped pixels)--integers 1-100 only: ');
input_image = imread(image_name, image_type);
%Read input image
A = double(input_image);
%Converts A to double precision matrix for computations
k = 128;
[m,n] = size(A); %Allows images of varying sizes and
%will allow FOR loop to pass through all pixels in input image.
%---------------------------------------------------
%---------------------------------------------------
%Percentage function: Asks user for percentage of pixels to test and performs %necessary calculations for 'for' loops to follow.Three while loops require percent %variable to be an integer between 1 and 100, inclusive.
while ((isa(begin_percent, 'int8')==0) &((begin_percent < 1) | (begin_percent > 100)))
%Requires input to be an integer
begin_percent = input('Invalid entry. Enter percentage of pixels to test--integers 1-100 only: ');
end
while ((isa(end_percent, 'int8')==0) & ((end_percent < 1) | (end_percent > 100)) & ((begin_percent +end_percent) > 100) & (begin_percent >end_percent))
%Requires input to be an integer
end_percent = input('Invalid entry. Enter percentage of pixels to test--integers 1-100
only: ');
end
begin_percent = double(begin_percent)/100;
%Converts percent variable to float point so the percentage is in decimal form, i.e., %10%=0.10
end_percent = double(end_percent)/100 + begin_percent;
%Converts percent variable to float point so the percentage is in decimal form, i.e., %10%=0.10
percent = abs(end_percent - begin_percent);
%---------------------------------------------------
%---------------------------------------------------
total_pixels = floor(percent*m*n);
%Calculates total number of pixels to test
skip_pixels = floor(begin_percent*m*n);
skip_rows = floor(begin_percent*m);
begin_row = skip_rows + 1;
begin_col = skip_pixels - skip_rows*n + 1;
second_row = begin_row + 1;
last_whole_row = floor(end_percent*m);
%Calculates # of rows to test necessary number of total_pixels
```

EK-3 (Devam) Matlab Code

```
%(for use in the for loop)
end_pixel = floor(end_percent*m*n);
last_col = end_pixel - last_whole_row*n;
%---------------------------------------------------------
%---------------------------------------------------------
X=zeros(k,1);
%Initialize 128x1 vector to zero. X will count frequencies of even-valued colors.
Y=zeros(k,1);
%Initialize 128x1 vector to zero. Y will count frequencies of odd-valued colors.
D=zeros(k,1);
%Initialize 128x1 vector to zero. D will assist in calculation of Chi-squared statistic.
%Following loop counts the frequencies of color values in pixels of input image. If %the pixel value is even, i.e., == 2i, then the (i+1)th row of X is incremented. If the %pixel value is odd, i.e., == 2i+1, then the (i+1)th row of Y is incremented. The %(i+1)th row is used because the vectors X and Y are by default indexed from 1 to %128.
for q=begin_row
for r=begin_col:n
if rem(A(q,r),2)==0
X((A(q,r)/2)+1)=X((A(q,r)/2)+1)+1;
else
Y(((A(q,r)-1)/2)+1)=Y(((A(q,r)-1)/2)+1)+1;
end
end
end
for q=second_row:last_whole_row
for r=1:n
if rem(A(q,r),2)==0
X((A(q,r)/2)+1)=X((A(q,r)/2)+1)+1;
else
Y(((A(q,r)-1)/2)+1)=Y(((A(q,r)-1)/2)+1)+1;
end
end
end
if last_whole_row < m
for q=last_whole_row + 1
for r=1:last_col
if rem(A(q,r),2)==0
X((A(q,r)/2)+1)=X((A(q,r)/2)+1)+1;
else
Y(((A(q,r)-1)/2)+1)=Y(((A(q,r)-1)/2)+1)+1;
end
end
end
end
%---------------------------------------------------------
%---------------------------------------------------------
for i=1:128
if (X(i)+Y(i)) < 5
X(i) = 0;
Y(i) = 0;
k = k-1;
end
end
Z = (X + Y);
```

EK-3 (Devam) Matlab Code

```
Z=Z./2;
%Vector of the averages of X_i and Y_i: Z_i =(X_i + Y_i)/2.
D = (X-Z).^2;
for i=1:128
if Z(i)==0
D(i) = 0;
else
D(i) = D(i)/Z(i);
end
end
format long g;
C=sum(D);
%C is the Chi-squared statistic, the sum of the elements in D.
display(C);
%Outputs C to the screen.
display(k);
%Begin debugging code
%display(m);
%display(n);
%display(percent);
%display(total_pixels);
%display(whole_rows);
%display(columns_last_row);
%End debugging code
%P=1 - (quadl(@integrand,0,C,1.e-15));
%Calculates probability that image has an embedded message.
Pcheck=1-gammainc(C/2,(k-1)/2);
%Checks value calculated for P using gammainc function.
%display(P);
%Outputs probability P to the screen.
display(Pcheck);
%Outputs Pcheck to screen.
```

EK-3 (Devam) Matlab Code

3.5 DÇ3.m

```
%DÇ3
image_name = input('Enter image name and location (exclude extension): ', 's'); %Location and name of image to be analyzed
image_type = input('Enter image type, e.g., png, bmp:', 's');
%Input image type
input_image = imread(image_name, image_type);
%Read input image
A = double(input_image);
%Converts A to double precision matrix for computations
[m,n] = size(A); %Allows images of varying sizes and
%will allow FOR loop to pass through all pixels in input image.
percentage = zeros(101,1);
prob = zeros(101,1);
K = zeros(101,1);
%-----------------------------------------------------
%----------------Tests all percentages, 1-100---------
for h=1:100
k = 128;
percentage(h) = percentage(h) + h;
%Converts percent variable to float point so the percentage is in decimal form, i.e.,
%10%=0.10
total_pixels = floor((h/100)*m*n);
%Calculates total number of pixels to test
whole_rows = floor(total_pixels/n);
%Calculates # of rows to test necessary number of total_pixels
%(for use in the for loop)
columns_last_row = total_pixels - (whole_rows*n);
%Calculates # of columns (elements) in last (possibly partial) row to test necessary %number of total_pixels (for use in the for loop)
X=zeros(128,1);
%Initialize 128x1 vector to zero. X will count frequencies of even- valued colors.
Y=zeros(128,1);
%Initialize 128x1 vector to zero. Y will count frequencies of odd-valued colors.
D=zeros(128,1);
%Initialize 128x1 vector to zero. D will assist in calculation of Chi-squared statistic.
%Following loop counts the frequencies of color values in pixels of input image.
%If the pixel value is even, i.e., == 2i,
%then the (i+1)th row of X is incremented.
%If the pixel value is odd, i.e., == 2i+1,
%then the (i+1)th row of Y is incremented.
%The (i+1)th row is used because the vectors X and Y are by default indexed from 1 %to 128.
for q=1:whole_rows
for r=1:n
if rem(A(q,r),2)==0
X((A(q,r)/2)+1)=X((A(q,r)/2)+1)+1;
else
Y(((A(q,r)-1)/2)+1)=Y(((A(q,r)-1)/2)+1)+1;
end
end
end
for q=whole_rows+1
```

EK-3 (Devam) Matlab Code

```
for r=1:columns_last_row
if rem(A(q,r),2)==0
X((A(q,r)/2)+1)=X((A(q,r)/2)+1)+1;
else
Y(((A(q,r)-1)/2)+1)=Y(((A(q,r)-1)/2)+1)+1;
end
end
end
Z = (X + Y);
for i=1:128
if (X(i)+Y(i)) < 5
X(i) = 0;
Y(i) = 0;
k = k-1;
end
end
Z=Z./2;
%Vector of the averages of X_i and Y_i: Z_i = (X_i + Y_i)/2.
D = (X-Z).^2;
for i=1:128
if Z(i)==0
D(i) = 0;
else
D(i) = D(i)/Z(i);
end
end
format long g;
C=sum(D);
%C is the Chi-squared statistic,the sum of the elements in D.
Pcheck=1-gammainc(C/2,(k-1)/2);
%Calculates probability using gammainc function.
%display(Pcheck);
prob(h) = prob(h) + Pcheck;
K(h+1) = K(h+1) + k;
end
fig_title = [image_name, '.', image_type,':','Probability of Embedding vs. Percentage Tested'];
percentage = cat(1, percentage(101),percentage(1:100));
prob = cat(1, prob(101), prob(1:100));
result = cat(2, percentage, prob, K);
display(result);
save('result','result');
figure (1), plot(percentage,prob,'*:');
title(fig_title); xlabel('Percentage Tested'); ylabel('Probability of Embedding'); %displays graph
%figure (2),
imshow(A, []); title('Tested Image');
%displays original image
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı	: HASSAN, Muhammad
Uyruğu	: Irak
Doğum tarihi ve yeri	: 21.02.1975 Kerkük
Medeni hali	: Bekar
Telefon	: 009 05434171812
e-mail	: m_75h@yahoo.com.

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Musul Üniversitesi /Fizik Bölümü	1999
Lisans	Kerkük Teknik Üniversitesi/ Bilgisayar Mühendisliği Bölümü	2004
Lise	Al-Velid Lisesi	1994

İş Deneyimi

Yıl	Yer	Görev
2007-2008	Gazi Üniversitesi	Gönüllü Araştırma Görevlisi

Yabancı Dil

İngilizce, Arabca

Hobiler

Tenis, Bilgisayar teknolojileri, Basketbol