ORTAM VE YÖNTEM BAĞIMSIZ STEGANOGRAFİK KÜTÜPHANE TASARIMI

Onur ÜNLÜ

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ŞUBAT 2012

ANKARA

Onur ÜNLÜ tarafından hazırlanan "ORTAM VE YÖNTEM BAĞIMSIZ STEGANOGRAFİK KÜTÜPHANE TASARIMI" adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Hacer KARACAN …….…….
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. Sezai DİNÇER …….…….
Elektrik-Elektronik Mühendisliği Anabilim Dalı, G.Ü.

Yrd. Doç. Dr. Hacer KARACAN …….…….
Bilgisayar Mühendisliği Anabilim Dalı, G.Ü.

Yrd. Doç. Dr. Hasan Şakir BİLGE …….…….
Bilgisayar Mühendisliği Anabilim Dalı, G.Ü.

Tarih: 06/02/2012

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

Prof. Dr. Bilal TOKLU …….…….
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.

Onur ÜNLÜ

ORTAM VE YÖNTEM BAĞIMSIZ STEGANOGRAFİK KÜTÜPHANE TASARIMI
(Yüksek Lisans Tezi)

Onur ÜNLÜ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Şubat 2012

ÖZET

Bilgi güvenliğine gereksinim her geçen gün artmaktadır. Bu gereksinimi karşılamada steganografik yöntemler önemli bir yere sahiptir. Steganografi temel olarak, bir varlığın bir yere saklanmasını ve istenen kimseler tarafından saklanan yerden çıkarılmasını hedefler. Üçüncü şahısların saklı varlığı algılayamaması ana amaçtır. Toprağa gömülmüş define buna örnek olarak verilebilir. Elektronik ortamda steganografi, bir bit dizesinin içerisine başka bir bit dizesini saklamaktır.

Bu çalışmanın amacı elektronik ortamda gerçekleştirilebilecek steganografi çeşitlerini bir kütüphanede toplamaktır. Bu bağlamda ortamdan ve yöntemden bağımsız steganografik bir kütüphane tasarlanmış ve bu kütüphaneyi kullanan bir uygulama geliştirilmiştir. Kullanılan tasarım desenleriyle kütüphanede esneklik sağlanmış böylece yeni yöntem ve ortamların kolayca eklenebilmesine imkan verilmiştir. Bu kütüphane ile ses, görüntü ve video dosyaları üzerinde saklama işlemi yapılabilmektedir. Yöntem olarak LSB (Least Significant Bit), LSBM (Least Significant Bit Matching) ve LSD (Least Significant Digit) yöntemleri desteklenmektedir.

Literatürde yapılan çalışmaların büyük çoğunluğu saklama ortamı olarak ses ve görüntü dosyalarını kullanmaktadırlar. Bu ortamlar ise boyutları gereği

saklama kapasitesini sınırlamaktadırlar. Bu çalışmayı diğerlerinden ayıran özellik video dosyaları üzerinde saklama yaparak kapasite kısıtını aşması ve video dosyasının görüntü ve ses alanını birlikte kullanarak güvenliği bir üst seviyeye çıkarmasıdır. Ayrıca çalışma kapsamında web kamerasından alınan videoya gerçek zamanlı olarak saklama gerçekleştirilmiştir. Böylece gizli nesnenin boyutuna göre video üretilmesi sağlanmıştır. Aynı zamanda orijinal video ile karşılaştırarak steganografinin varlığını tespit etme ihtimali ortadan kaldırılmıştır.

Yapılan saklama işlemlerinin başarımı çeşitli steganaliz ve gürültü ölçme yöntemleriyle değerlendirilmiş ve bunların sonuçları verilmiştir.

Bilim Kodu : 902.1.011
Anahtar Kelimeler : Steganografi, Veri gizleme, Veri gömme, LSB
Sayfa Adedi : 71
Tez Yöneticisi : Yrd. Doç. Dr. Hacer KARACAN

DESIGN OF A MEDIA AND METHOD INDEPENDENT STEGANOGRAPHIC LIBRARY
(M.Sc. Thesis)

Onur ÜNLÜ

GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
February 2012

ABSTRACT

Demand on information security increases day by day. To meet this requirement, steganographic techniques have a considerable place. Basically steganogoraphy aims hiding an entity into a place and its extraction by desired people. Undetectability of hidden entity by third parties is the main objective. Buried treasure could be an example for this. Electronic steganography is hiding a bit stream into another bit stream.

Objective of this work is to gather some methods of steganography in a library that would be achieved at the electronic medium in a library. In this context a steganographic library independent from medium and method is designed and an application which uses this library is developed. By using design patterns, flexibility in the library is achieved, thus easily adding new techniques and mediums becomes possible. It can hide data on sound, image and video files. LSB, LSBM and LSD methods are supported.

Most of the studies in the literature utilizes audio and image files as hiding medium. Because of their capacity, these medias limit hiding capacity. This work exceeds this limit as it can hide on video files. The key point that distinguishes this work from others is that it uses both image and audio domain together in video file so it takes security to an upper level. Also real time hiding

into video taken from webcam is implemented within the scope of this work. Thus video production according to the size of secret file is provided. Meanwhile possibility of detecting the existence of steganography by comparing with original video has been suppressed.

The success of hiding process is evaluated with various steganalysis and noise measurement methods and the results have been given.

Science Code : 902.1.011
Key Words : Steganography, Data hiding, Data embedding, LSB
Page Number : 71
Adviser : Assist. Prof. Dr. Hacer KARACAN

TEŞEKKÜR

Her zaman yanımda olan aileme, çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren Hocam Yrd. Doç. Dr. Hacer KARACAN'a teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÖZET....iv

ABSTRACT....vi

TEŞEKKÜR....viii

SİMGELER VE KISALTMALAR....xiv

1. GİRİŞ....1

2. STEGANOGRAFİ....4

3. STEGANOGRAFİDE KULLANILAN ORTAMLAR....7

3.1. Ses Dosya Formatları....7

3.1.1 WAV....8

3.1.2 MP3....8

3.2. Görüntü Dosya Formatları....9

3.2.1. BMP....10

3.2.2. GIF....10

3.2.3. PNG....11

3.2.4. JPEG....11

3.3. Video Dosya Formatları....14

3.3.1. AVI....14

3.3.2. MPEG....15

4. LİTERATÜRDE STEGANOGRAFİK ÇALIŞMALAR....17

4.1. Metin Steganografi....18

4.2. Ses Steganografi....19

4.3. Görüntü Steganografi....20

Sayfa

4.3.1. Görüntünün uzaysal alanını kullanan yöntemler.....20

4.3.2. Görüntünün frekans alanını kullanan yöntemler.....30

4.4. Video Steganografi.....36

4.4.1. AVI dosyalarında steganografi.....37

4.4.2. MPEG dosyalarında steganografi.....38

4.4.3. Mevcut video steganografik yazılımlar.....38

5. GELİŞTİRİLEN UYGULAMA (StegoFolderLock).....42

5.1. Kütüphanede Kullanılan Teknolojiler.....43

5.2. Kütüphanenin Tasarımı ve Çalışma Mantığı.....44

6. DENEYLER.....52

7. SONUÇLAR VE ÖNERİLER.....63

KAYNAKLAR.....65

ÖZGEÇMİŞ.....71

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 4.1. Gizli bit seçme haritası.....19

Çizelge 4.2. LSB saklama işlemi esnasında son bit değişimi.....21

Çizelge 4.3. LSD saklama işlemi esnasında son bitlerin değişimi.....27

Çizelge 4.4. MSU StegoVideo, OpenPuff ve StegoFolderLock'un karşılaştırması....41

Çizelge 6.1. Uçak PSNR ve MSE değerleri.....57

Çizelge 6.2. Renkler PSNR ve MSE değerleri.....58

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 2.1. Steganografinin çalışma mantığı....4

Şekil 2.2. Sihirli Üçgen....5

Şekil 2.3. Cardan ızgarası yöntemiyle gizli mesaj örneği....6

Şekil 3.1. PCM örneği....7

Şekil 3.2. MP3 sıkıştırma işlemi....9

Şekil 3.3. Görüntü dosyalarının yapısı....10

Şekil 3.4. Orijinal görüntü ve ayrıştırılması....12

Şekil 3.5. JPEG blok dönüşüm adımları....13

Şekil 3.6. Dijital görüntü ve üç kere DWT uygulanmış görüntü....14

Şekil 3.8. GOP örneği....16

Şekil 4.1. Görüntü dosyasının formatından faydalanarak veri saklama....17

Şekil 4.2. Çince karakterler....19

Şekil 4.3. Görsel saldırı örneği....22

Şekil 4.4. Ki-kare saldırısı....22

Şekil 4.5. Pseudo Random indekslere saklama işlemi....23

Şekil 4.6. b-f) %10'dan %50'ye kadar saklama yapılmış görüntünün LSB hattı....25

Şekil 4.7. Spiral LSB yöntemi....26

Şekil 4.8. MMS için PNG ile steganografi....29

Şekil 4.9. JPEG Dosyalarında Frekans Alanına Veri Gömme İşlemi....30

Şekil 4.10. Veri gizlemeden önce ve sonra DCT katsayılarının histogramı....31

Şekil 4.11. Ardışık ve dağıtık saklama örnekleri....32

Şekil 4.12. Histogram karşılaştırması....33

Şekil **Sayfa**

Şekil 4.13. QIM, HM ve F5 histogram karşılaştırması.....33

Şekil 4.14. Ayrık dalgacık dönüşümü bantların gösterimi.....35

Şekil 4.15. Ardışık üç çerçeve ve renk bileşen histogramları.....37

Şekil 4.16. OpenPuff yazılımı arayüzü.....40

Şekil 4.17. MSU StegoVideo yazılımı arayüzü.....40

Şekil 5.1. Stegofolderlock dosya gizleme arayüzü.....42

Şekil 5.2. Stegofolderlock gizli dosyaları çıkarma arayüzü.....43

Şekil 5.3. Gereksinimler kullanım durumu diyagramı.....44

Şekil 5.4. Sistem genel sınıf diyagramı.....45

Şekil 5.5. Gizleme işlemi ardışıl diyagramı.....46

Şekil 5.6. Çıkarma işlemi ardışıl diyagramı.....46

Şekil 5.7. Giz paketinin sınıf diyagramı.....47

Şekil 5.8. Giz okuma işlemi ardışıl diyagramı.....48

Şekil 5.9. Ortu paketi sınıf diyagramı.....49

Şekil 5.10. Yontem paketi sınıf diyagramı.....50

Şekil 6.2. Orijinal ve saklama yapılmış ses dosyalarının spektogramları.....54

Şekil 6.3. Uçak görüntüsüne görsel saldırılar.....55

Şekil 6.4. Uçak görüntüsü, LSB, LSBM ve LSD'sine istatistiksel saldırı.....56

Şekil 6.5. Uçak görüntüsüne göre daha karışık olan görüntü.....57

Şekil 6.6. Renkler görüntüsü ve LSB, LSD' sine istatistiksel saldırı.....58

Şekil 6.7. Orijinal ve saklama yapılmış videolardaki seslerin dalga biçimleri.....59

Şekil 6.8. Orijinal ve saklama yapılmış videolardaki seslerin spektogramları.....60

Şekil **Sayfa**

Şekil 6.9. Videolardan alınmış kesit (Yukarıdan aşağı ardışık çerçeveler).....61

Şekil 6.10. Videolardan alınmış kesit üzerinde steganaliz.....62

Şekil 7.1. Güvenliği arttırmak için uygulanan saklama yöntemi.....63

SİMGELER VE KISALTMALAR

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

Kısaltmalar	Açıklama
AVI	Audio Video Interleave
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
J2ME	Java 2 Mobile Edition
JPEG	Joint Photographic Experts Group
GIF	Graphics Interchange Format
GOP	Group of Pictures
LSB	Least Significant Bit
LSBM	Least Significant Bit Matching
LSBMR	Least Significant Bit Matching Revisited
LSD	Least Significant Digit
MDCT	Modified Discrete Cosine Transform
MMS	Multimedia Messaging Service
MPEG	Motion Pictures Experts Group
MP3	MPEG Layer III
PCM	Pulse Code Modulation
PNG	Portable Network Graphics
PRNG	Psuedo Random Number Generator
RIFF	Resource Interchange File Format
SLSB	Spiral Least Significant Bit
SSIS	Spread Spectrum Image Steganography
WAV	Waveform Audio File Format

1. GİRİŞ

Günümüzde insanlar arasındaki bilgi paylaşımı hızla artmaktadır. Bu durum çeşitli güvenlik sıkıntılarını da beraberinde getirmektedir. Güvenlik açıklarına her gün yenileri eklenmektedir. Bilgi iletişiminde güvenliği sağlamanın iki ana yolu vardır. Bunlar Kriptoloji ve Steganografidir.

Kriptoloji gönderilecek bilginin şifrelenmesine ve alıcı tarafta deşifre edilmesine dayanır. Böylece iletişim ortamını dinleyen üçüncü şahıslar bilgiyi elde edebilmek için şifreyi kırmak zorundadırlar. Burada önemli olan şifreli varlığın üçüncü şahısların eline geçmemesi değil, eline geçtiğinde dahi içeriğinin anlaşılamamasıdır. Kriptoloji gücünü şifreleme algoritmasından alır. Algoritma çözüldüğünde sistem kırılmış sayılır.

Steganografi anlamlı bir varlığın (örtü nesne) içine daha küçük boyutlu anlamlı bir varlığı (gizli nesne) gizleme tekniğidir. Gizli bilgiyi, masum görünen bir varlığın içinde gizleyerek göndermeye dayanır. Burada önemli olan masum görünen varlığın üçüncü şahısları veya yazılımları şüphelendirmemesidir. Masum varlığın gizli bilgi taşıdığı anlaşıldığında sistem kırılmış sayılır.

Gerçek dünyada hazinenin belirli bir yere gömülmesi ve sonra harita yardımıyla bulunup çıkartılması steganografiye örnek olabilir. Sanal dünyada ise gizli nesne anlamlı bir bit topluluğu, örtü nesne ise uygun bir dosya türü olabilir.

Steganografi bilginin saklandığı ortama göre çeşitlenir. Bu ortam html, exe gibi herhangi bir dosya formatı olabilir. Günümüzdeki yöntemler ağırlıkla görüntü, ses ve video dosyalarını kullanmaktadır.

Tez kapsamında yapılan çalışmanın amacı ortam ve yöntem bağımsız, esnek yapıya sahip steganografik bir kütüphane tasarlamaktır. Böylece yeni yöntem ve ortamlar kolayca eklenebilir. Kütüphanenin, ses ve görüntü dosyalarındaki saklama kapasitesini aşabilmek için video dosyalarına saklama yapmaya izin vermesi

elzemdir. Aynı zamanda dahili kamera aracılığı ile gizlenecek nesnenin ihtiyaç duyduğu boyutta videoyu çekerek gerçek zamanlı bilgi saklamaya izin vermesi güvenliği arttırıcı bir unsurdur.

Gerçekleştirilen kütüphane kullanılarak WAV formatındaki ses dosyalarına, BMP formatındaki görüntü dosyalarına ve AVI (1.0) formatındaki video dosyalarına veri saklama yapılabilmektedir. Bu dosya türleri sıkıştırma kullanılmayan ham dosya türleridir. Bunların tercih edilmesindeki ana sebep sıkıştırma işleminin karmaşıklığından uzak olmalarıdır. Saklanan varlık dosya sistemindeki bir dosya veya klasör olabilir.

Saklama işleminde LSB, LSBM ve LSD yöntemleri desteklenmektedir. LSB yöntemi elektronik ortamda uygulanan steganografik yöntemlerin temelidir denilebilir. LSBM yöntemi istatistiksel saldırılara karşı LSB yöntemine kıyasla daha başarılı bir yöntemdir. LSD yöntemi ise steganografik yöntemlerde aranan yüksek kapasite düşük algılanabilirlik kriterlerini sağlamaktadır.

Jhonson, steganografik yöntemleri ortamdan bağımsız bir şekilde sınıflandırarak inceleyen detaylı bir çalışma yapmıştır [1]. Cheddad ise görüntü dosyalarında uygulanabilecek steganografik yöntemleri inceleyen ve karşılaştıran kapsamlı bir çalışma gerçekleştirmiştir [2]. Lashkari'nin görüntü steganografide kullanılan algoritmaları değerlendiren bir makalesi de mevcuttur [3].

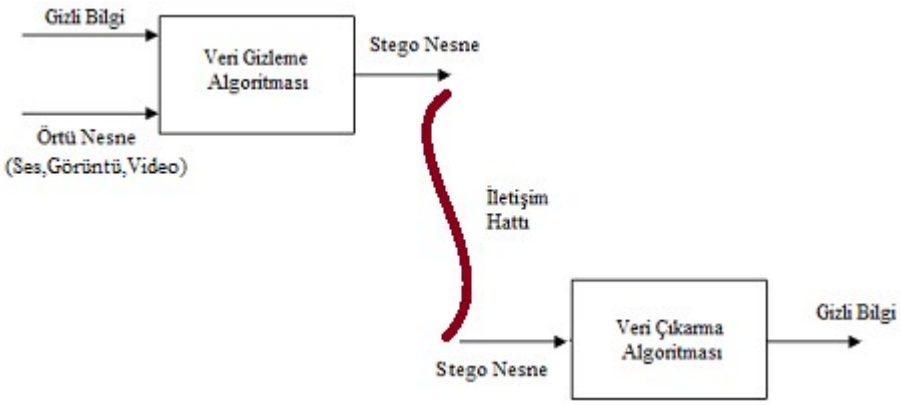
Literatürde daha önce benzer olarak Zekeriya Erkin ve Bülent Örencik'in tasarladığı kütüphane mevcuttur [71]. Bu kütüphane; yer değiştirmeye, frekans alanında katsayı değiştirmeye, yayılmış spektrum kullanımına dayanan çeşitli yöntemlerle görüntü ortamında bilgi saklamaya izin vermektedir. Çalışmalarında yöntemlerin saklama kapasitesine etkilerini, görüntü üzerindeki değişimi ve çeşitli görüntü işleme tekniklerine karşı dayanıklılıklarını incelemişlerdir.

Bir sonraki bölümde steganografi, başarım ölçütleri ve geçmişten günümüze gelişimi anlatılacaktır. Üçüncü bölümde elektronik platformda steganografinin uygulandığı

ortamlar ve bu ortamların özelliklerinden bahsedilecektir. Dördüncü bölümde literatürde metin, ses, görüntü ve video steganografisi üzerine yapılmış çalışmalar incelenmiştir. Beşinci bölümde geliştirilen kütüphane ve uygulama, altıncı bölümde yapılan deneyler anlatılmıştır. Son bölümde elde edilen sonuçlar ve yapılabilecek potansiyel çalışmalar verilmiştir.

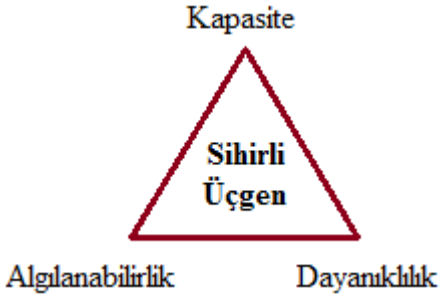
2. STEGANOGRAFİ

Steganografi gizli iletişim bilimidir. Kelime olarak steganografi, "Örtülü Yazı" anlamına gelen Yunan sözcüklerinden türetilmiştir [2]. Temel fikir önemli veriyi sıradan örtü nesne (metin, hareketsiz görüntü, ses, hareketli görüntü) içine gömerek stego nesneyi oluşturmaktır. Stego nesne örtü nesne ile aynı yöntemle gönderilir ve depolanır. Habersiz gözlemciler için stego nesne, örtü nesnenin içeriğini sergiler. Hedeflenen alıcıların bakış açısından stego nesne örtü nesnenin kamufle ettiği önemli bilgiyi taşır [4]. İşlem genel olarak üç aşamadan oluşur. Gizli bilgi örtü nesne içerisine veri gizleme algoritması kullanılarak saklanır böylece stego nesne oluşmuş olur. Stego nesne iletişim hattı üzerinden alıcıya gönderilir. Alıcı stego nesneden veri çıkarma algoritmasıyla gizli bilgiyi elde eder. Bu işlem Şekil 2.1'de gösterilmektedir.



Şekil 2.1. Steganografinin çalışma mantığı

Steganografik yöntemler için algılanabilirlik, dayanıklılık ve kapasite faktörleri başarı kriterleri olarak gösterilebilir. Genel olarak bunlar birbirlerini olumsuz etkileyen faktörlerdir. Bu üç faktör sihirli üçgen olarak da adlandırılmaktadır (Şekil 2.2) [5]. Algılanabilirlik, ortamda gizli veri taşınıp taşınmadığının üçüncü şahıslar tarafından sezgilerle veya çeşitli araçlarla algılanamamasıdır. Dayanıklılık, stego nesne üzerinde yapılacak kesme, gürültü, süzgeç uygulama gibi nesneyi değiştiren işlemler sonucunda gizli verinin hala elde edilebilir olmasıdır. Kapasite ise ortama saklayabileceğimiz en fazla veri miktarını belirtir. Ses dosyalarında bu üç faktörün birbiriyle olan ilişkisini inceleyen bir çalışma literatürde mevcuttur [6].

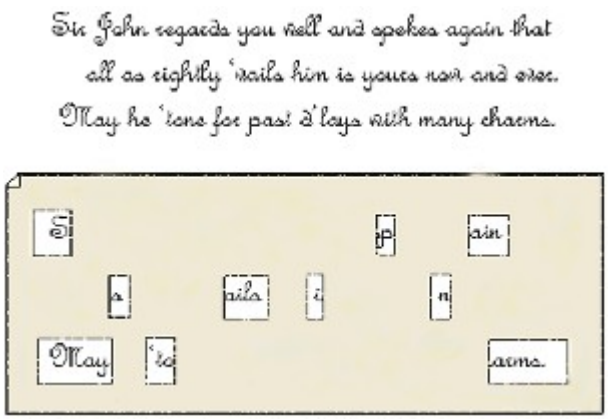


Şekil 2.2. Sihirli Üçgen

Steganaliz, gizli verinin (Steganografik yöntemle saklanmış veri) mevcudiyetini tespit etme bilim ve sanatıdır [7]. Odak noktası gizli mesajın varlığını açığa çıkarmak ve mümkünse içeriğini elde etmektir. Genel olarak bir algoritma verilen nesnenin steganografik bilgi içerip içermediğini tahmin edebiliyorsa, steganografik sistem o algoritma tarafından kırılmış sayılır [8].

Günümüzde internet üzerinden pek çok dosya paylaşımı yapılmaktadır. Bunların çoğunluğunu görüntü, ses ve video dosyaları oluşturmaktadır. Bunların masum görünen dosyaların içerisinde gizli bilgi bulunma olasılığı göz ardı edilemeyecek bir gerçektir. Bu fikir üzerinden giderek internetteki görüntü dosyaları üzerinde steganaliz uygulayan bir çalışma [9] da yapılmıştır.

Steganografinin ilk kullanımlarının M.Ö. 1500 yıllarına dayandığı söylenmektedir [10]. M.Ö. 5. yüzyılda Histaiacus kölesinin saçlarını kestirmiş ve kafasına gizli mesajı dövme olarak yaptırmıştır. Saçları uzadıktan sonra köleyi gizli mesajın ulaşacağı yere göndermiştir. 500 yıl önce İtalyan Matematikçi Jerome Cardan, antik Çin'den kalma gizli yazı yöntemini ortaya attı. Üzerinde delikler bulunan maske ve mesaj birlikte kullanılarak gizli metin elde ediliyordu. Bu yönteme Cardan ızgarası denmektedir (Şekil 2.3) [2]. Görünmez mürekkep tekniği II. Dünya Savaşında çokça kullanılan bir yöntemdir. Bu yöntemde gizli mesaj kağıt üzerine kimyasal bir madde ile yazılır. Mesajı okuyabilmek için kağıda kimyasal işlemler uygulamak gerekir [10].



Şekil 2.3. Cardan ızgarası yöntemiyle gizli mesaj örneği [2]

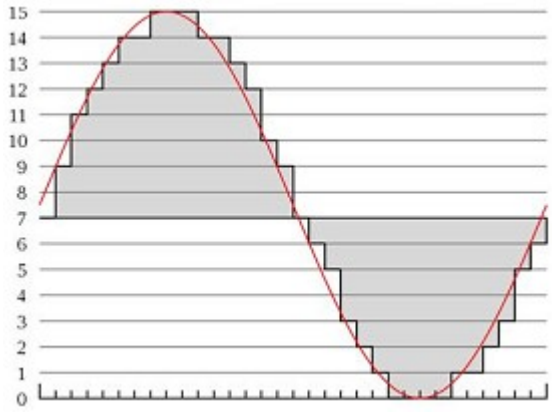
Günümüzde ise istihbarat birimleri ve terör örgütlerinin muhtemel iletişim aracı steganografidir. Bu tip uygulamaları bulmak ve kullanmak son derece kolaydır. İnternette Turksteg adında bit eşlem dosyalar içine veri gizleme ve çıkarma yapan Türkçe bir uygulama da mevcuttur [11]. Bunun yanı sıra Fujitsu firması görüntü çıktı alınmadan içerisine veri gizleyen ve çıktı alındıktan sonra cep telefonu ile fotoğrafı çekilip gönderildiğinde gizli bilgiyi çıkaran bir uygulama geliştirmiştir [2].

3. STEGANOGRAFİDE KULLANILAN ORTAMLAR

Steganografik yöntemleri anlayabilmek için bu yöntemlerin üzerinde kullanıldığı ortamları anlayabilmek gerekmektedir. Bu bölümde steganografinin kullanıldığı ortamlardan bahsedilecektir. Ses, görüntü ve video dosyalarının yapısı bunların nasıl oluşturuldukları ve nasıl sıkıştırıldıkları anlatılacaktır. Ses dosyalarından WAV, MP3, görüntü dosyalarından BMP, GIF, PNG, JPEG, video dosyalarından da AVI ve MPEG'e değinilecektir.

3.1. Ses Dosya Formatları

Ses dosyaları, ses dalgalarının sayısal ortamda tutulmalarını sağlarlar. Doğadaki ses sinyallerinin zamana göre belirli aralıklarla örneklenmesiyle oluşturulan sayısal dizi, PCM (Pulse Code Modulation) dizisidir (Şekil 3.1).



Şekil 3.1. PCM örneği [12]

PCM orijinal analog sinyale yaklaşmayı sağlar. Eğer analog sinyal 3.71 ölçülürse bu 4 olarak sayısallaştırılacaktır. Bu fark nicemleme hata miktarıdır. Bir ses örneğini ifade etmek için kullanılan bit sayısı arttıkça hata miktarı azalacaktır. Günümüzdeki uygulamalar bir ses örneği için 8 bit veya daha fazla rakam kullanmaktadır. Bu örnekleme değerleri insan kulağının nicemleme hatasını algılayamaması için yeterlidir.

Orijinal sesin tekrar üretilebilmesi için analog sesteki en yüksek frekans değerinin iki katı örnekleme yapılmalıdır. İnsan sesi genellikle 8 kHz ile örneklenir. Çoğu ses uygulaması insan kulağının bütün tayfını kapsayacak şekilde 44.1 kHz frekans kullanırlar [13].

Örneğin örnekleme için 8 bit kullanan bir ses uygulaması 44.1 kHz'de stereo ses üretecek olursa 44100 x 8 x 2 = 705600 bit, yani her saniye için yaklaşık 86 Kilobayt veri depolayacaktır.

Görülmektedir ki ham ses dosyaları kapladığı yer açısından oldukça büyüktür ve bilgisayarda depolamak için çok uygun değildir. Yer kısıtını ortadan kaldırmak amacıyla çeşitli sıkıştırma yöntemleri geliştirilmiştir. Bunlar kayıplı ve kayıpsız sıkıştırma olmak üzere ikiye ayrılır. Kayıplı sıkıştırma yöntemleri kayıpsızlara göre daha fazla sıkıştırma sağlamaktadır. Bu yöntemler genel olarak insan işitme sisteminin algılayamayacağı ses bilgilerinin atılmasına dayanmaktadırlar. Kayıpsız sıkıştırma formatlarının başında MP3 gelmektedir. Bu bölümde ham veri formatlarından WAV ve sıkıştırılmış veri formatlarından MP3'den bahsedilecektir.

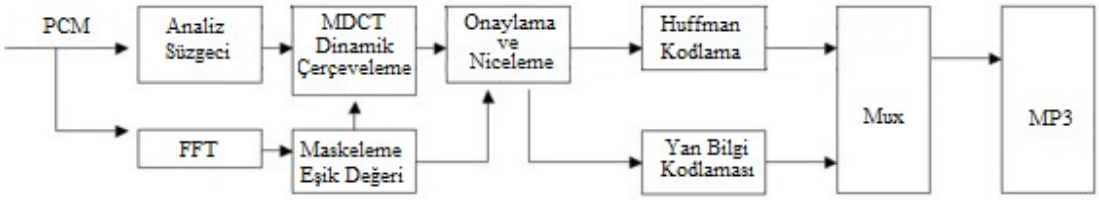
3.1.1 WAV

Microsoft tarafından geliştirilmiş RIFF (Resource Interchange File Format) dosya standardının bir alt türüdür. WAV (Waveform Audio File Format) dosya formatı örnekleme frekansı ve örnek derinliği gibi tanımlayıcı bilgileri içeren veri yığınları ve ham PCM dizilerini içeren veri yığınlarından oluşur. 4 Gigabayta kadar veri saklama kapasitesi vardır. Ham ses verilerinin yanı sıra sıkıştırılmış verilere de desteği vardır.

3.1.2 MP3

MP3 (MPEG Audio Layer III) olarak adlandırılan format 1987'de Almanya'da geliştirilmeye başlanmıştır. 4 yıllık bir çalışmanın ardından orijinal sesin 1/12 boyuta kadar sıkıştırma başarılmıştır [14].

MP3 sıkıştırma işlemi karmaşık bir işlemdir. Birkaç aşamadan meydana gelmektedir. İlk olarak ham PCM verisi çok aşamalı süzgeçten geçirilir böylece ses eşit mesafeli alt frekans bantlarına ayrılmış olur. Bu örneklerden kritik olanlar seçilip sıkıştırılır. Bu seçim işleminde çıkabilecek bir problem alt bantların üstüne gelen kritik frekanslardan dolayı seste gürültüler oluşmasıdır. Bunu önlemek için her alt bandın menzili komşu alt bantlara genişletilir ve MDCT (Değişmiş Ayrık Kosinüs Dönüşümü) uygulanır [14]. İnsan işitme sisteminin algılamayacağı şekilde maskeleme uygulanır. Nicemleme ve kodlama işlemleriyle maskeleme gereksinimlerini karşılayarak veriyi sıkıştırır [15]. Sıkıştırma işlemi genel olarak Şekil 3.2'de görülmektedir.

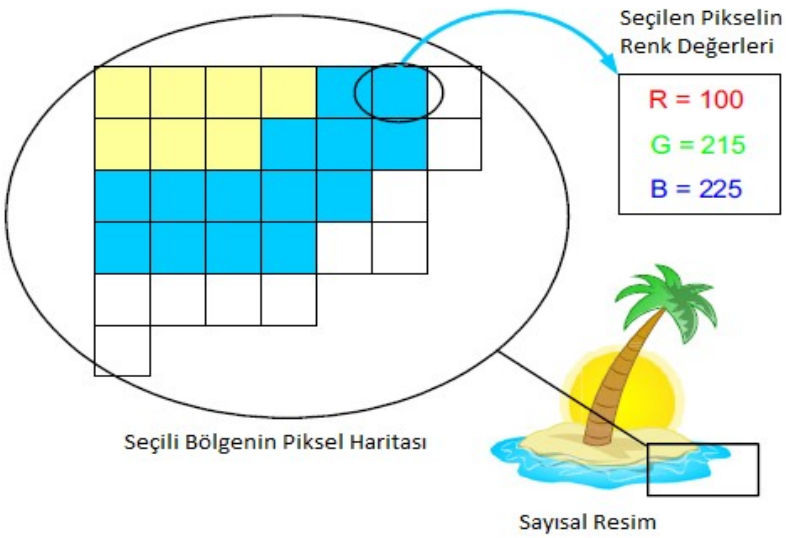


Şekil 3.2. MP3 sıkıştırma işlemi [14]

3.2. Görüntü Dosya Formatları

Görüntü dosyaları başlık ve görüntü olmak üzere iki kısımdan oluşur. Başlık kısmında görüntüye ait genişlik, yükseklik, boyut gibi bilgiler bulunur. Görüntü m genişlik ve n yükseklik olmak üzere m x n renk bilgisi içeren noktadan (piksel) oluşur. Renkli görüntüler için her bir noktada kırmızı, yeşil ve mavi üç renk bandını ifade eden bilgi mevcuttur (Şekil 3.3). Her renk bitlerle ifade edilir ve bir rengi ifade eden bit sayısı arttıkça görüntünün kalitesi ve boyutu artar.

Görüntü dosya türlerinden en yaygın kullanılanlar BMP, GIF (Graphics Interchange Format), PNG (Portable Network Graphics), JPEG (Joint Photographic Experts Group)'dir.



Şekil 3.3. Görüntü dosyalarının yapısı [16]

3.2.1. BMP

BMP dosya türü en basit görüntü dosya türüdür denilebilir. Bu formatta sıkıştırma kullanılmaz. Bu yüzden dosya boyutları büyüktür ve internet üzerinden transfer için uygun değildir. Bunun en çok kullanılan türü 24 bit BMP dosyasıdır. Bu formatta her renk bandı 8 bit ile ifade edilir. Bir piksel kırmızı, yeşil ve mavi bant için 8 bit olmak üzere toplam 24 bitten oluşur. Bu durumda 1 mega piksellik bir 24 bit BMP dosyanın boyutu yaklaşık 3 mega bayt olacaktır. BMP dosyaları veri gizleme için uygundur. Örneğin 24 bit BMP dosyada kırmızı, yeşil ve mavi banttaki 8 bitin son bitleri veri gizleme için kullanılabilir.

3.2.2. GIF

GIF dosya türü görüntüyü sıkıştırarak kaydeder. Bu dosya türünün çalışma mekanizması şu şekildedir: Görüntüdeki renkler 16 milyon renkten 256 renge ölçeklenir. Görüntü içerisinde ölçekleme bilgisini içeren bir indeks tablosu bulunur. Görüntü açılırken bu tablo yardımı ile açılır. GIF formatı kayıpsız format olarak geçmektedir ancak bu görüntü içerisinde 256 veya daha az farklı renk mevcut ise doğrudur. Eğer görüntüde 256' dan fazla farklı renk var ise orijinal görüntü kayıplı

olarak GIF formatında sıkıştırılabilir [17].

3.2.3. PNG

PNG dosya türü görüntüyü sıkıştırarak kaydeden diğer bir dosya türüdür. Bu dosya türünde sıkıştırma yöntemine satır bazlı karar verilir. İndeks tablosu ve Huffman yöntemini de kullanan bu formatın özelliği görüntüyü gerçek anlamda kayıpsız sıkıştırmasıdır. İnternet üzerinde desteklenen ve gerçek manada kayıpsız sıkıştırma yapan tek dosya türüdür. PNG formatında da bilgi saklanıp kayıpsız bir şekilde çıkartılabilir. Ancak bu durumda orijinal PNG görüntüyle saklama yapılmış PNG görüntü arasında boyut farkı olacaktır.

3.2.4. JPEG

JPEG dosyaları kayıplı sıkıştırma yaptığı için ve görüntü renklerinin gizleme için direk kullanımına izin vermediği için ilk başlarda kaçınılmıştır. JPEG üzerinde çalışma yapılmaya başlandıktan sonra veri gizleme açısından daha sağlam ve saldırılara karşı daha dayanıklı olduğu görülmüştür [2].

Günümüzde internette en çok kullanılan format kaliteli yüksek oranlı sıkıştırma sağlayan JPEG'dir. Bu dosya türünde veri gizleme için ayrık kosinüs dönüşümü (DCT) katsayılarından faydalanılır. Veri gizlemenin nasıl yapılacağının anlaşılması için önce sıkıştırma mantığının anlaşılması gerekir.

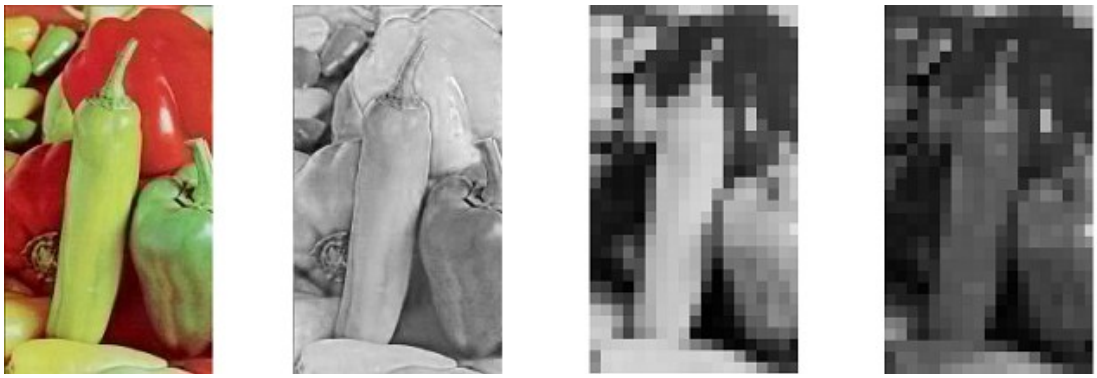
JPEG dosya türünün sıkıştırma tekniği diğer görüntü formatlarından daha karmaşıktır. JPEG kayıplı sıkıştırma yapar yani sıkıştırma işleminden sonra orijinal görüntü geri elde edilemez. Bu format, gözün rengin detaylarından daha çok parlaklığını algılamasından faydalanır.

İlk olarak RGB değerleri Yoğunluk(Y), Mavi - Sarı(Cb) ve Kırmızı - Yeşil(Cr) değerlerine çevrilir. Bu çevirme işleminde çoğu zaman 4 komşu piksele bir değer atanır. RGB-YCbCr çevirme işleminde kullanılan formül aşağıdaki gibidir [18].

$$Y = (0.257 * R) + (0.504 * G) + (0.098 * B) + 16$$

$$Cr = (0.439 * R) - (0.368 * G) - (0.071 * B) + 128$$

$$Cb = -(0.148 * R) - (0.291 * G) + (0.439 * B) + 128$$



RGB görüntü Yoğunluk (Y) Mavi – Sarı (Cb) Kırmızı– Yeşil(Cr)

Şekil 3.4. Orijinal görüntü ve ayrıştırılması [19]

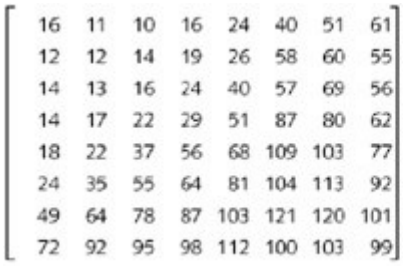
Bu işlemden sonra görüntü 8x8'lik bloklar halinde küçük görüntülere ayrılır ve her görüntüde DCT işlemi uygulanır. Aşağıda 8x8'lik ilk piksel değerlerini içeren matris ve DCT işleminden sonraki matris görülmektedir. Bu esnada görüntü eski haline kayıpsız döndürülebilir. Bu adımdan sonra 8x8'lik blok JPEG tarafından önerilen nicemleme tablosundaki değerlerle birebir bölünür ve sayılar yuvarlanır. Bilgi kaybı bu esnada yaşanmaktadır. Bu işlem görüntüde belirgin olmayan bir yumuşamaya sebep olacaktır. Nicemleme tablosundaki değerler arttıkça sıkıştırma oranı ve görüntüdeki bozulma artmaktadır. Bu işlemden sonra sonuç bloktaki sayıların çoğu birbiriyle aynı hale gelmiştir. Bunları kullanarak sıkıştırma yapılır ve Huffman ağacı ile JPEG son halini alır [19].

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

a) Piksel değerleri

1260	-1	-12	-5	2	-2	-3	1
-23	-17	-6	-3	-3	0	0	1
-11	-9	-2	2	0	-1	-1	0
-7	-2	0	1	1	0	0	0
-1	-1	1	2	0	-1	1	1
2	0	2	0	-1	1	1	-1
-1	0	0	-1	0	2	1	-1
-3	2	-4	-2	2	1	-1	0

b) DCT'den sonraki piksel değerleri



16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99



79	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

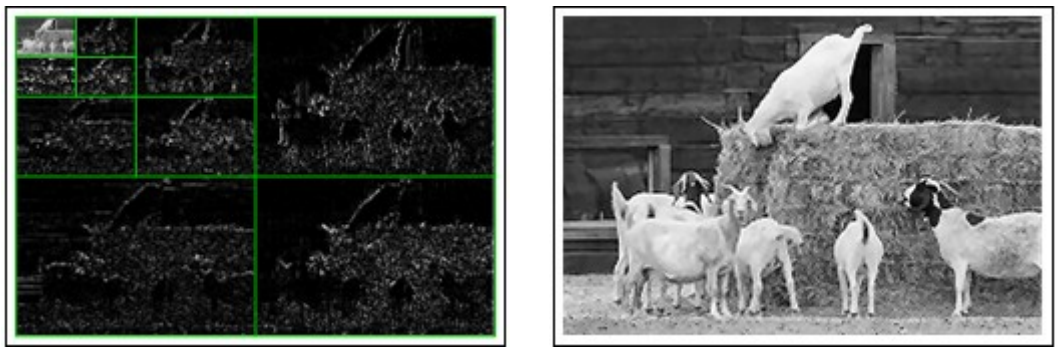
c) Nicemleme tablosu d) Bölünüp yuvarlama yapılmış değerler

Şekil 3.5. JPEG blok dönüşüm adımları [19]

JPEG 2000

2000 yılında yeni jenerasyon JPEG olarak tasarlanan ve mevcut versiyonun yerine geçmesi amacıyla çıkan JPEG 2000, DCT'den farklı bir dönüşüm olan DWT (Discrete Wavelet Transform)'yi kullanmaktadır. Dosya uzantısı jp2'dir.

Dalgacıklar, belirli özellikleri sağlayan ve bir fonksiyon gösterimini diğerine dönüştürmek amaçlı kullanılabilen matematiksel fonksiyonlardır. Dalgacık dönüşümü çoklu çözünürlüklü görüntü analizi yapar. Çoklu çözünürlük analizi, görüntünün eş zamanlı olarak birden fazla çözünürlük seviyelerinde gösterimidir. Dalgacık dönüşümü, görüntüyü farklı yerlerde ve farklı ölçeklere sahip dalgacık fonksiyonlarının toplamı olarak ifade eder. Görüntünün dalgacıklara ayrıştırılması işlemi dalga form çiftlerini gerektirmektedir. Biri görüntünün detaylı kısımlarını içeren yüksek frekansları içerirken diğeri görüntünün düz kısımlarını içeren düşük frekansları içerir [20].



Şekil 3.6. Dijital görüntü ve üç kere DWT uygulanmış görüntü [21]

DWT işlemi tamamlandıktan sonra oluşan görüntüdeki her bloka ayrı ayrı nicemleme işlemi uygulanır [21]. Son olarak da kodlanır.

3.3. Video Dosya Formatları

Video dosyaları bir küme hareketsiz görüntünün belirli bir hızda arka arkaya oynatılmasıyla oluşur. Bu esnada ilişkili ses bilgisi görüntü dosyasıyla senkron bir şekilde çalar. Videoların çeşitli karakteristikleri vardır. Bunlar çerçeve boyutu, fps(saniyedeki çerçeve sayısı), çerçevedeki pikselin renk bileşenleri gibidir. İnsan gözünün bir görüntüyü hareket olarak algılayabilmesi için video dosyasının saniyede 24 çerçeve civarında fps hızına sahip olması gerekir.

Video dosya türleri sıkıştırmasız ve sıkıştırmalı olmak üzere ikiye ayrılır. Sıkıştırmasız türlerine örnek AVI (Audio Video Interleave) dosya türüdür. Sıkıştırmalı dosya türlerine örnek olarak da MPEG (Moving Picture Experts Group) verilebilir. Bu bölümde bu iki formattan bahsedilecektir.

3.3.1. AVI

Microsoft ve IBM tarafından RIFF geliştirilen AVI temel olarak bit eşlem görüntülerinin arka arkaya oynatılmasına dayanır. Hiçbir sıkıştırma kullanılmadığı için dosya boyutları oldukça büyüktür. Kullanılan ses dosyası ise farklı olabilir.

3.3.2. MPEG

MPEG sıkıştırma kullanarak kodlama yapan video dosya formatıdır. Bu format ardışık çerçeveler arasındaki piksellerdeki değişimlerin az olmasından faydalanır. Denizde giden bir geminin videosunu düşünürsek ardışık iki çerçeve arasında gemi çok hafif bir yer değiştirme yapacak ve piksellerin çoğu aynı kalacaktır.

MPEG'de üç tip çerçeve vardır. Bunlar I (Intra-coded), P (Predicted-coded) ve B (Bidirectionally predicted coded) tipi çerçevelerdir. P ve B tipi çerçevelere, iki I tipi çerçevenin arasında yer aldıkları için ara kodlanmış (inter-coded) çerçeve de denilmektedir. Bunlar I tipi çerçeveden sonraki değişiklikleri tutarlar.

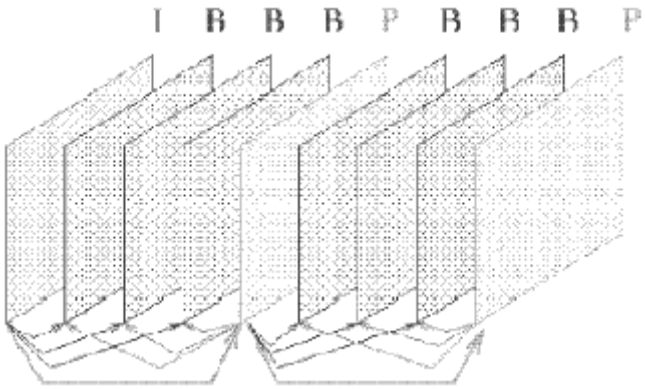
Bu çerçeve türlerinden en önemlisi başlı başına çerçevenin tamamını tutan I tipi çerçevelerdir. Bu çerçeveler JPEG benzeri sıkıştırma yöntemi kullanılarak sıkıştırılır. Bu çerçeveler başka çerçeveye ihtiyaç duymaksızın görüntüyü oluşturabilmektedir.

P tipi bir çerçeve görüntüyü oluşturabilmek için kendinden önceki ilk I veya P tipi çerçeveye ihtiyaç duymaktadır. Bu çerçeve kodlanırken önceki ilk I veya P çerçevesinden elde edilen görüntüdeki her makro blokun aktif görüntüdeki yeri bulunur. Yeri bulunan makro blok artık hareket vektörü olarak kodlanabilir. Ancak her zaman tam olarak eşleşme gerçekleşmemektedir. Eşleşmeyen pikseller tahmin hatası olarak sıkıştırılarak kodlanır. Eşleşme oranı yüksek oldukça bu kodlama yöntemi makro bloku ham olarak kodlamaktan daha tasarruflu olacaktır. Eşleşme oranının düşük olduğu durumlarda ise ham makro blok olarak kodlamak daha iyi sonuç verecektir. Bir P tipi çerçeve, bir I tipi çerçevenin yaklaşık yarı boyutundadır.

B tipi çerçeve kodlanırken ise kendinden önceki ilk I veya P ve kendinden sonraki ilk I veya P tipi çerçeveyi birlikte kullanarak kendi görüntüsünü oluşturur. Bir B tipi çerçeve, bir I tipi çerçevenin yaklaşık çeyreği boyutundadır.

Videodan videoya değişmekle beraber tipik olarak 1 I çerçevesinden sonra 14 civarı P veya B tipi çerçeve kullanılır. Örneğin bir sahnenin değiştiğini düşünürsek P veya

B çerçevesi yerine I çerçevesi kullanmak daha verimli bir çözüm olacaktır. Bir I çerçeve ve ardındaki P ve B çerçevelerin tamamına GOP (Group Of Pictures) denir.



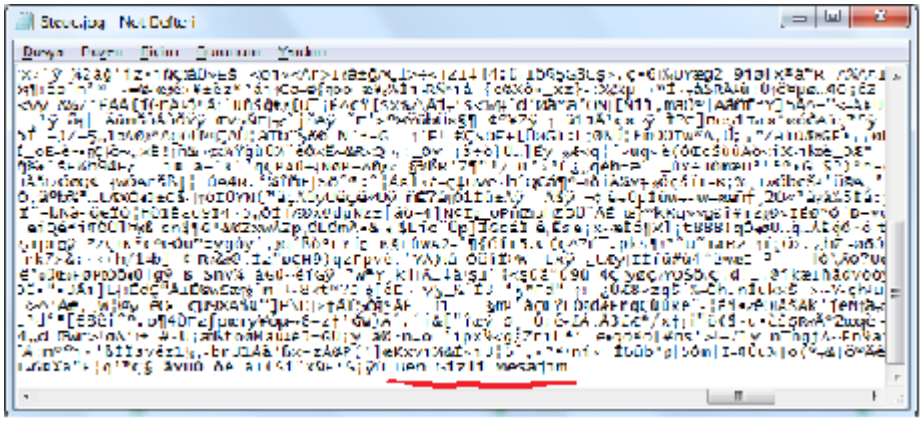
Şekil 3.8. GOP örneği [22]

4. LİTERATÜRDE STEGANOGRAFİK ÇALIŞMALAR

Steganografi dijital ortamda hemen her yerde kullanılabilir. Sayısal ortamda yapılan saklama 3. şahıslar tarafından fark edilemediği sürece yöntemimiz başarılıdır. Bu anlamda en temel yöntemler dosyaların başlık, etiket, renk gibi dosyaya has özelliklerini kullanarak uygulanan yöntemlerdir.

Görüntü dosyaları bilgi saklama açısından en çok kullanılan ortamlardan biridir. Görüntü dosyalarının başlık ve görüntü bölümlerine bilgi gizlenebilir. Başlık alanı gibi görüntünün formatından faydalanılarak bilgi gizleme yapılırsa dosya boyutu değişecektir.

Örneğin görüntü dosyalarının başlık kısmı veya dosya bitimindeki EOF (dosyanın sonu) etiketinden sonraki kısımlar veri gizlemek için kullanılabilir [2]. “C:>copy “Student Identity.jpeg” /b + Gizli.txt /b Stego.jpeg” gibi basit bir Windows işletim sistemi komutu kullanılarak steganografi gerçekleştirilebilir.



Şekil 4.1. Görüntü dosyasının formatından faydalanarak veri saklama

Bu işlemin sonucunda Stego.jpeg dosyasını Not Defteri ile açtığımızda en alt kısımda gizli bilgi eklenmiş olarak görüyoruz (Şekil 4.1). Bu işlem Exif gibi başlık formatları kullanılarak da gerçekleştirilebilir. Veriyi gizlemeden önce bir şifreleme işleminden geçirmek güvenliği artırmak için uygun olacaktır.

Bu tür saklama yapan örneklerin sayısını arttırmak mümkündür. Mehmet Ali Atıcı tezinde xml, html, exe ve word dosyalarında bu dosyaların çeşitli karakteristiklerini kullanarak steganografi gerçekleştiren uygulamalardan bahsetmiştir [23].

Bu bölümde literatürde kullanılan steganografik yöntemler uygulandıkları ortamlara göre sınıflandırılarak incelenecektir.

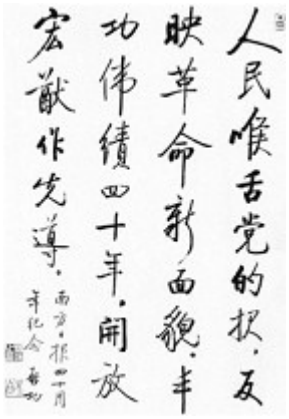
4.1. Metin Steganografi

Metin steganografi en eski steganografik yöntemlerden biridir. Tarihimizden buna örnek olarak cümlelerin içerisine tarih saklama yöntemi olan ebcet hesabı verilebilir. Metin steganografi, steganografinin en zor uygulandığı alandır [24]. Bu ses ve görüntü gibi diğer ortamlara nazaran tekrar eden lüzumsuz verinin çok daha az olmasından kaynaklanmaktadır. Bu yöntemi Tayland dilinde, İngilizce, Japonca, Korece, Çince, Farsça ve Arapça'da başarıyla uygulayan çalışmalar bulunmaktadır [25].

Shirali-Shahreza bir çalışmasında [24] Arapça ve Farsça'daki harflerin çoğunda noktalama işaretlerinin bulunduğunundan bahsetmiş ve bu işaretlerin dikey yüksekliklerinin saklama amacıyla kullanıldığı bir yöntem önermiştir.

Shirali-Shahreza bir diğer çalışmasında [26] Amerikan İngilizcesi ve İngiliz İngilizcesi'nde aynı anlamı ifade eden farklı kelimeleri değiştirerek veri saklayan bir çalışma yapmıştır. Örneğin İngiliz İngilizcesindeki “football” sözcüğü Amerikan İngilizcesinde “soccer” olarak karşılık bulmaktadır. Bu yöntemin düşük saklama kapasitesine sahip olduğu ancak İngilizce bilgisi düşük olan ülkelerde gizli iletişim için uygun olduğu belirtilmiştir.

Bir çalışmada [27], Çince'deki basitleştirilmiş ve geleneksel karakterlerin birbirinin yerine kullanılabildiğinden bahsedilmiş ve bu özelliğin steganografi amacıyla kullanılabileceği belirtilmiştir. 1 için geleneksel karakterlerin, 0 için ise basitleştirilmiş karakterlerin kullanılması önerilmiştir.



Şekil 4.2. Çince karakterler [27]

Türkçe metinlere gizli bilgi yükleme olanaklarını değerlendiren bir çalışma da yapılmıştır [28]. Bu çalışmada yapısal, anlamsal ve işaretsel değişiklikler incelenmiştir.

4.2. Ses Steganografi

Pakistan'dan üç akademisyen çalışmalarında LSB yönteminin gelişmiş bir versiyonunu önermektedirler [29]. Bu yöntemde saklama işleminde 8 bitlik ses örnekleri kullanılmakta, örnek ve saklama yapılacak bit seçimi bir haritaya göre yapılmaktadır. Böylece üçüncü şahıslara saklamanın rastgele yapıldığı izlenimi verilmektedir. Bu da istatistiksel saldırılara karşı güvenlik seviyesinin artmasını sağlanmaktadır.

Çizelge 4.1. Gizli bit seçme haritası

İlk İki Bit	**Gizli Bit İndeksi**
00	3. LSB
01	2. LSB
01/11	1. LSB

Bit seçiminde rastgelelik ses örneğinin ilk iki bitine göre karar verilerek

sağlanmaktır. Hangi durumda hangi indeksin kullanılacağı Çizelge 4.1'de gösterilmiştir. Ses örneği seçiminde de benzer yöntem izlenmiş, bir ses örneğinin ilk 3 biti kendinden sonraki saklama yapılmış ses örneğini işaret etmektedir. Ortalama saklama kapasitesi 4 ses örneğine 1 bittir.

Diqun Yan ve Rangding Wang 2009 yılında yayınlanan makalelerinde MP3 sıkıştırma standardının özelliklerinden biri olan Huffman kodlaması işleminde tablo seçimi yaparak steganografi sağlamışlardır [30]. Diğer yöntemlere kıyasla yüksek kapasite sağlayan aynı zamanda güvenlik ve düşük bozukluk gereksinimlerini de karşılayan bir yöntemdir. MP3 standardı Huffman kodlaması için 34 farklı tablo sunmaktadır. Kodlama öncesinde bu tablolardan en iyi sıkıştırma sağlayan bir kaçı tercih edilebilir. Bu tabloları iki gruba ayırarak grupların gizli bilgiyi içermesi sağlanmıştır.

Shirali-Shahreza, konuşma seslerindeki sessizlik aralıklarının uzunluklarını değiştirerek gerçek zamanlı saklama yapan ve MP3 sıkıştırmaya dayanıklı yeni bir yöntem önermiştir [31].

4.3. Görüntü Steganografi

Görüntü dosyalarında bilgi gizleme temel olarak iki ana başlıkta toplanabilir bunlar uzaysal alan kullanılarak gizleme ve frekans alanı kullanılarak gizlemedir.

4.3.1. Görüntünün uzaysal alanını kullanan yöntemler

Görüntünün uzaysal alanını kullanan yöntemler sıkıştırma kullanılmayan görüntülerde kullanılır. Bu yöntemlerin temel hedefi görüntü uzayında yapılan değişikliklerin görüntü üzerinde asgari görsel ve istatistiksel değişikliğe sebep olmalıdır.

LSB Yöntemi

Steganografi görüntü dosyalarında genel olarak LSB yöntemini kullanır. Bu yöntemde görüntünün bir pikselini oluşturan bitlerden rengi en az değiştirecek bit değiştirilir. Böylece kendi gizli bilgimizi yerleştirebileceğimiz alanı elde etmiş oluruz. Örneğin bir pikselin 8 bitten oluştuğunu düşünelim: 11010010 (210). Kullanıcı için bu piksel bir rengi ifade eder. Bu pikselin son bitinde değişiklik yapacak olursak: 11010011 (211). Bu değişiklik görüntü üzerinde, kullanıcının dikkatle baksa bile anlayamayacağı bir etki yapacaktır. Bu durumda saklama kapasitemiz 8 bite bir bit olduğu için toplamda dosya boyutunun yaklaşık 1/8' i kadar bilgiyi gizlememiz mümkündür. Bilgi gizleme kapasitesini en az önemli n biti kullanarak n/8' e çıkartabiliriz ancak n arttıkça renk üzerindeki değişiklik miktarı artacak ve görüntüdeki bozukluk fark edilir hale gelecektir.

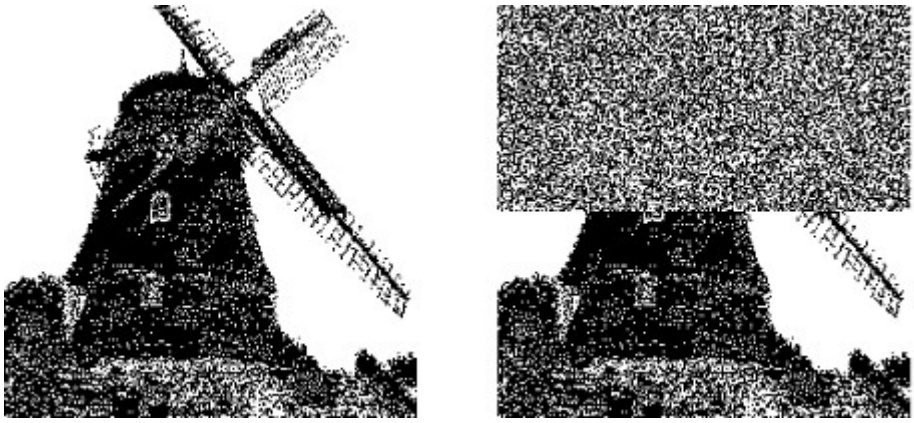
Çizelge 4.2. LSB saklama işlemi esnasında son bit değişimi

Saklamadan Önce Örtü Görüntü Bayt Değeri	**Saklanacak Bit**	**Saklamadan Sonra Örtü Görüntü Bayt Değeri**
1011111**1** (191)	1	1011111**1** (191)
1111001**1** (231)	1	1111001**1** (231)
1000011**1** (135)	0	1000011**0** (134)

LSB Yöntemine Saldırılar

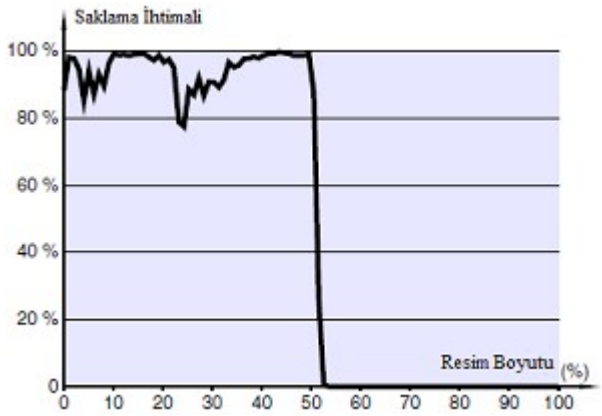
Andres Westfeld ve Andreas Pfitzmann 2000 yılında yayınladıkları makalede standart LSB yöntemiyle saklama yapıldığında tespit edebilen görsel ve istatistiksel saldırı yöntemlerini sunmuşlardır [32].

Bu saldırı yöntemi 8 bitten oluşan renk değerinin ilk 7 bitini sıfırlamakta böylece görüntünün renk değerlerinin sadece 0 ve 1'lerden oluşur hale getirmektedir. Sonra 1 değerleri en büyük renk değeri olan 255 dönüştürülmekte, 0 değerleri ise olduğu gibi bırakılmaktadır. Bu işlem basit LSB kullanılan görüntüler için uygun olabilmektedir. Ancak daha karışık bir görüntü seçildiğinde bu yöntem sonuç vermemektedir. Şekil 4.3'de LSB'lere göre siyah beyaz değer alan görüntüler görülmektedir. Bu görüntülerden ikincisinin yarısına kadar veri saklandığı anlaşılmaktadır.



Şekil 4.3. Görsel saldırı örneği [32]

Son bitlerde LSB yöntemiyle değişiklik yapıldığında görüntü üzerinde istenmeyen etkiler meydana gelmektedir. Son bitlerdeki 1 ve 0'ların mevcutlarının birbirine oranı orijinal görüntüde yüksek farka sahip iken, saklama yapılmış görüntüde birbirine yakın olmaktadır. Bu istatistiksel farkı kullanan saldırıya ki-kare (chi-square) saldırısı denir (Şekil 4.4).



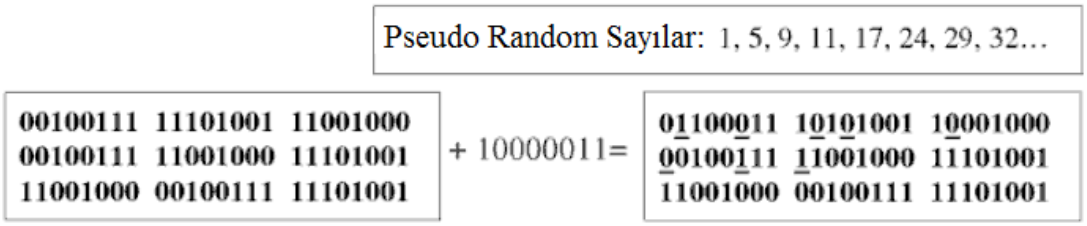
Şekil 4.4. Ki-kare saldırısı [32]

Burada uygulanan ki-kare saldırısına daha dayanıklı görüntüler elde edilmesi için saklama yapılacak piksel seçiminde PRNG (Pseudo Random Number Generator) kullanılabilir. PRNG kullanımının dezavantajı alıcı ile gönderici arasında bir ön haberleşmeyi zorunlu kılmasıdır. Ayrıca PRNG kullanılırsa güvenlik açısından

gizlenecek bilginin boyutu düşük tutulmalıdır.

İstatistiksel saldırıların görsel saldırılara olan avantajı orijinal görüntüye ihtiyaç duymaksızın kullanılabilmesi ve otomatize edilerek çok sayıda görüntüye hızlı bir şekilde uygulanabilmesidir.

Steganografiyi tespit etmeye yönelik bu saldırı yöntemlerinin yanı sıra makalelerinde Guillermo A. Francia ve Tyler S. Gomez steganografi yok ediciyi önermişlerdir. Burada amaç steganografiyi ne tespit etmek ne de gizli veriyi elde etmektir. Amaç steganografik bilgiyi yok etmek böylece gizli iletişimi önlemektir. Geliştirdikleri yazılımda son görüntü içerisindeki son 4 bite kadar temizlemeye izin vermektedir [33].



Şekil 4.5. Pseudo Random indekslere saklama işlemi [3]

Pseudo Random (Sahte Rastgele) Yöntemler

Bu yöntem LSB yönteminin düzenli bir şekilde saklama yapmasından kaynaklanan istatistiksel tahribatı azaltmak amacıyla kullanılmıştır. PRNG kullanımı iletim yapan tarafların bir ön anahtar paylaşımını zorunlu kılmaktadır. PRNG tarafından aynı sayının tekrar üretildiği durumda çakışma meydana gelmektedir. Bu durum kullanılmış bitlerin indekslerinin bir listede tutulmasıyla önlenebilir. Aynı yöntem çıkarma işleminde de kullanılmalıdır. Bu saklama işleminde indeks seçiminde görüntüdeki tüm bitler kullanılacak olursa istenmeyen gürültülere sebep olacaktır (Şekil 4.5) [3].

Patchwork (Yama işi) Yöntemi

Patchwork, sahte rastgele ve istatistiksel işlem tabanlı bir yaklaşımdır. Bu yöntemde Gauss dağılımına sahip bir istatistik, örtü görüntüye gömülür [34]. Bu algoritmada iki ana adım vardır:

i) Sahte rastgele yöntemle iki yama seç
ii) İlk yamaya d sabit sayısını ekle ikinci yamadan d sabit sayısını çıkar.

Parlaklık değerleri üzerinde uygulanan arttırma ve azaltma aynı miktarda yapılır. Bu sebeple ortalama parlaklık değeri değişmez.

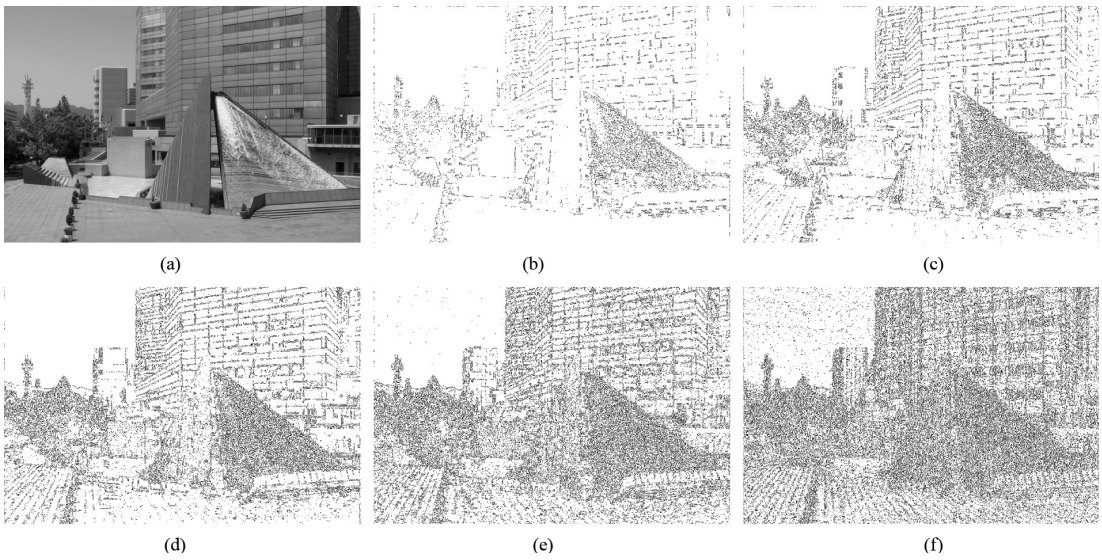
2003 yılında Kore'den iki akademisyen bu yöntemin gelişmiş bir versiyonunu ses dosyaları üzerinde uygulamışlardır [35]. Aynı araştırmacılar bir diğer makalelerinde frekans alanında saklama yapan arttırımsal patchwork ve çarpansal patchwork algoritmalarını birlikte kullanan yöntemi sunmuşlardır. Yöntemin sıkıştırma ve sinyal işleme saldırılarına karşı etkili olduğunu söylemişlerdir [36].

LSBM Yöntemi

İlk defa Toby Sharp tarafından önerilen LSBM [37] yöntemi, son biti değiştirme yöntemiyle standart LSB yönteminden ayrılmaktadır. Bu yöntemde örtüdeki son bit gizlenecek bitle değiştirilmez. Eğer örtüdeki son bit gizlenecek bitten farklı ise 1 arttırmak veya 1 azaltmak şeklinde değiştirilerek saklama yapılır. Bu yöntem frekans alanında kullanıldığında stego görüntü, istatistiksel saldırılara karşı daha dayanıklı hale getirilmiştir.

Jarno Mielikainen 2006 yılında bu yöntemi daha da geliştirmiş ve aynı miktarda veriyi daha az pikseli değiştirerek saklamıştır. Önerdiği yöntemde pikselleri çift olarak kullanmış ve birinci pikselin son bitine bir bitlik bilgi saklamış, piksel çiftinin bir fonksiyonundan ise ikinci, bir bitlik bilgiyi elde etmiştir. Bu sayede örtü görüntü saldırılara karşı daha dayanıklı hale gelmiştir [38].

Çin'den üç akademisyen 2010 yılında yayınlanan makalelerinde Jarno Mielikainen önerdiği yöntemi daha da geliştirmişlerdir. Örtü görüntüdeki düz ve pürüzsüz alanların veri saklama işlemi sonrasında görsel olarak zayıf olacağını ve düşük güvenlik sağlayacağını söylemişlerdir. Bunu önlemek için keskinliğe duyarlı olarak saklama bölgelerini seçebilen, gizli mesajın boyutuna ve ardışık iki pikselin farklarına göre saklama yapan yöntemlerini önermişlerdir. Yöntem, saklanacak bilgi küçük olduğunda örtü görüntüdeki düz alanları olduğu gibi bırakarak keskinliğin yoğun olduğu yerlere saklama yapmaktadır. Yöntem güvenliği gözle görülür şekilde arttırmaktadır [39].



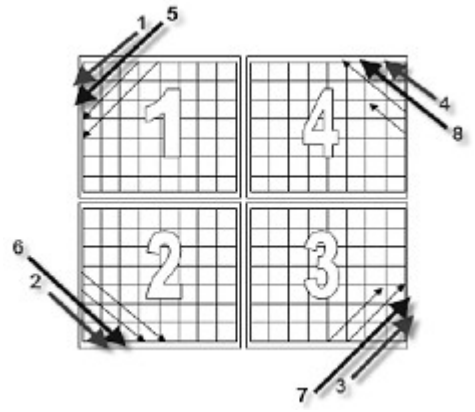
Şekil 4.6. b-f) %10'dan %50'ye kadar saklama yapılmış görüntünün LSB hattı [39]

Şekil 4.6'da önerilen yöntemin düşük saklama oranlarında düz alanlara saklama yaptığı görülmektedir. Yöntem, saklama oranı arttıkça düz bölgelere de saklama yapmaktadır bu da güvenliği düşürmektedir.

SLSB (Spiral Least Significant Bit) Yöntemi

King Saud üniversitesinden 4 araştırmacı çalışmalarında farklı bir yöntem önermişlerdir. Bu yöntemde görüntü n adet eşit bloğa bölünmüş (örneğin 100 x 100'lük görüntüyü 50 x 50'lik 4 bloka bölme) ve spiral olarak LSB uygulanmıştır.

Blokların dış köşesinden veya merkezlerinden veya karma olmak üzere üç türlü yöntemi, saat yönünde veya saatin tersi yönünde uygulama seçeneği sunmuşlardır. Elde ettikleri sonuçlar bu yaklaşımın ki-kare saldırısına karşı oldukça dayanıklı olduğunu göstermiştir [40].



Şekil 4.7. Spiral LSB yöntemi [40]

LSD Yöntemi

Feyzi Akar 2004 yılında yayınlanan çalışmasında RGB değerlerini ağırlıklandırarak saklama yapan yöntemini önermiştir [41]. Bu yöntem ile düşük görsel bozulmayla yüksek saklama kapasitesi sağlanmıştır. Bu yöntem son bit yerine son basamağı değiştirmeye dayandığı için bu yönteme LSD (Least Significant Digit) yöntemi denilebilir. LSD yöntemi LSB'nin bir türevidir. LSB yönteminde ikili sayı sistemindeki son basamaklar kullanılırken LSD yönteminde onluk sayı sistemindeki son basamak saklama işlemi için kullanılır. Sayı on arttırılır ve saklanacak basamak çıkarılır. 250 ve üzerindeki sınır değerler saklama işlemi için kullanılmaz. Bu saklama yönteminde LSB'ye göre daha çok veri saklanabilirken saklamanın gözle anlaşılabilirliği artmaktadır. Saklama öncesi ve sonrasında sayı değeri arasındaki fark en fazla 9 olsa da ikilik sistemde bakıldığında üst bitleri değiştirebildiği görülmektedir (Çizelge 4.3). Bu durum görüntüyü sağlayan uygulamanın bit değerlerine göre hesaplama yapması durumunda görüntüde büyük bozulmalara sebebiyet verecektir.

Çizelge 4.3. LSD saklama işlemi esnasında son bitlerin değişimi

Saklamadan Önce Örtü Görüntü Bayt Değeri	Saklanacak Basamak(3 bit)	Saklamadan Sonra Örtü Görüntü Bayt Değeri
10111111 (191)	1	11000111 (190 + 10 – 1 = 199)
11110011 (231)	4	11111000 (230 + 10 – 4 = 236)
10000111 (135)	7	10000101 (130 + 10 – 7 = 133)

Blok Eşleme Yöntemi

Ran-Zan Wang ve Yeh-Shun Chen yayınlanan çalışmalarında çift yönlü blok eşleme yöntemini sunmuşlardır. Bu yeni yöntem gizli görüntü ile örtü görüntüyü blok olarak eşleştirmeye dayanmaktadır. İki görüntü m x n boyutunda bloklara ayrılır. Gizli görüntüdeki i. bloka örtü görüntüdeki en benzer blok bulunur ve eşleştirilir. Bu işlem sürekli tekrarlanır. İşlem sonucunda iyi eşleşmeyen bloklar ve eşleşen blokların indeks bilgileri sıkıştırılır ve örtü görüntüye dağılacak şekilde LSB'lere saklanır. Deney sonucunda düşük gürültü ve yüksek kapasite sağladığı yazarlar tarafından belirtilmiştir [4].

Shang-Kuan Chen ve Ran-Zan Wang 2010 yılında yayınlanan çalışmalarında mevcut blok eşleme yöntemini geliştirmişlerdir. Blok arama alanını genişletmişler ve böylece saklanacak bilgi miktarını düşürmüşlerdir. Böylece önceki çalışmaya göre daha kaliteli örtü ve gizli görüntü elde etmişlerdir [42].

Renk Paleti Tabanlı Yöntem

GIF dosya türü sıkıştırma öncesinde bilgi saklamak için çok uygun değildir çünkü 256'dan fazla renk olması durumunda bilgi kaybı olacaktır. Bu da LSB yöntemiyle yaptığımız değişikliğin kaybolması ve doğru şekilde geri alınamaması anlamına gelecektir. Sıkıştırma sonrasında saklama yapılacak olursa, paletteki renklere bağlı olarak görüntüde gözle görülür değişiklik olması ihtimali çok yüksektir. Bu durumda

paletteki komşu renk benzerliğine dayalı adaptif bir yöntem uygulanabilir.

GIF formatındaki indeks tablosu da bilgi gizleme için kullanılabilir ancak tablo 256 renk indeksinden oluştuğu için 256 bit yani 32 bayt bilgi saklanabilecektir. Bu ise çok küçüktür.

Son olarak sadece gri tonlu renklerden oluşan görüntülerin kullanılması önerilmiştir. Eğer renk paletindeki gri tonları yeterince çeşitliyse bu türlü görüntüler veri gizleme için en uygun GIF ortamlarıdır ve saklanan verinin gözle algılanması son derece zordur [43].

Spread Spectrum (Yayılmış Spektrum)

Yayılmış Spektrum iletişim dar banttaki sinyalin geniş bant sinyal üzerinde yayılarak gönderilmesidir. Bu yayılma işleminden sonra dar bant sinyali tespit etmek zorlaşır.

Lisa M. Marvel ve arkadaşları 1999 yılında SSIS (Spread Spectrum Image Steganography) yöntemini tanıtmışlardır [44]. Bu yöntem gizli bilgiyi gürültü içerisine saklamayı temel alır. Gizli bilgiyi başarıyla çıkarabilmek için restorasyon teknikleri kullanılmış. Restorasyon ile orijinal görüntü elde edilebilirse eklenen gürültü de elde edilecektir. Bu tekniklerle bilgi hatasız olarak elde edilemediği için saklamadan önce hata kontrol kodlaması uygulanmıştır. Ayrıca saklanan verinin görüntünün iyi seçilmiş parametrelerle sıkıştırılmasına dayanabildiği tespit edilmiştir.

Lisa'nın yöntemine kaotik şifreleme ve kaotik modülasyon ekleyen ve böylece daha yüksek güvenlik sağlayan bir çalışma da mevcuttur [45]. Bir çalışmada ise pikseldeki renk bayt'ının son biti yerine orta bitleri rastgele bir şekilde saklama amacıyla kullanılmış ve böylece sıkıştırma ve gürültü gibi etkilere daha dayanıklı ve daha güvenli hale getirilmiştir [46]. SSIS frekans alanında uygulandığında ise görsel olarak çok daha iyi performans göstermiş ve orta ve yüksek gürültü seviyelerinde daha da iyi sonuç vermiştir [47].

Diğer Yöntemler

Mohammed Shirali Shahreza, çoklu ortam mesajlaşma hizmeti (MMS) için geliştirdiği steganografik yöntemi anlatmıştır [48]. MMS mesajlarda metin (30 KB'ye kadar), görüntü (100 KB'ye kadar) ve video (300 KB'ye kadar) gönderilebilir.



Şekil 4.8. MMS için PNG ile steganografi [48]

Önerilen yöntemde mesajdaki metin ve görüntü alanı birlikte kullanılmış. Görüntüde gözün mavi renge daha az hassas olmasından ve metin alanında mesajlarda kullanılan kısaltma sözlerden (for/4, you/u) faydalanılmış. Kısaltma sözler 1 normal sözler 0 bit değerini ifade etmektedir. Metin alanına 1 bit görüntü alanına 5 bit saklanacak şekilde bir yol izlenmiş. Uygulama J2ME (Java 2 Mobile Edition) kullanılarak geliştirilmiş, görüntü formatı olarak PNG kullanılmış ve cep telefonlarında gizli iletişim için çok uygun olduğu görülmüş.

Seppanen, Makela ve Keskinarkus alıcı ve verici arasında bir anahtar ve palet paylaşımının zorunlu olduğu yüksek saklama kapasitesine sahip bir yöntem önermişlerdir. Palet, adaptif vektör nicemleme yöntemiyle oluşturulur. Palet daha sonra gizli bilginin çıkarılması için kullanılır. Bu yöntemle bir piksele yaklaşık 11 bite kadar saklama yapabilmişlerdir [49].

Brisbane ve arkadaşları 2005 yılında yayınlanan makalelerinde bu yöntemin

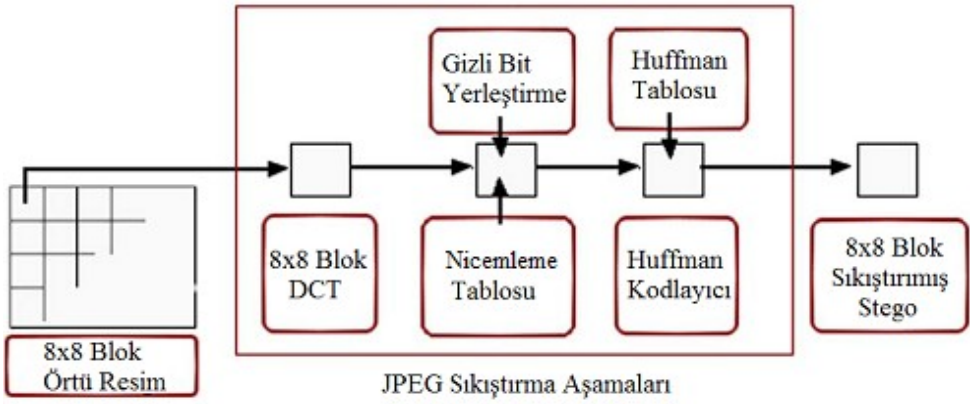
görüntüde yüksek seviye gürültüler oluşturduğunu ve bunun kabul edilemez sonuç verdiğini belirtmişlerdir. Bu yöntemi piksel seçiminde en az gürültüyü sağlayacak adaptif bir algoritma kullanarak iyileştirmişlerdir. Deneyler sonucunda 40 dB altında gürültüyle 6 bit/piksel saklama yaptığını görmüşlerdir [50].

4.3.2. Görüntünün frekans alanını kullanan yöntemler

Frekans alanından faydalanan yöntemler temel olarak iki ana başlıkta incelenebilir. Bunlar DCT ve DWT'dir.

DCT Steganografi

DCT steganografisinin en çok kullanıldığı görüntü dosyaları JPEG'dir. JPEG dosyalarında bir blok üzerinde DCT uygulanıp nicemleme ve yuvarlama yapıldıktan sonra elde edilen sonuç blok üzerinde LSB yöntemini uygularız. Bu işlem aşamalarıyla Şekil 4.9'da gösterilmektedir. Burada hangi değerlere bilgi saklanacağını iyi seçmek son derece önemlidir, çünkü bir değer üzerinde yapılan değişiklik tüm bloku etkileyecektir.



Şekil 4.9. JPEG Dosyalarında Frekans Alanına Veri Gömme İşlemi

Derek Upham 1997 yılında bu katsayılar üzerinde LSB kullanan J-Steg [51] yazılımını gerçekleştirmiştir. Kullanılan yöntem görsel saldırılara karşı dayanıklı olmasına rağmen istatistiksel saldırılara karşı zayıftır.

J-Steg Algoritması

Giriş: gizli Mesaj, Görüntü(JPEG)

Çıkış: stego Görüntü

gizli Mesajda saklanacak veri olduğu müddetçe

görüntüdeki bir sonraki DCT katsayısını al

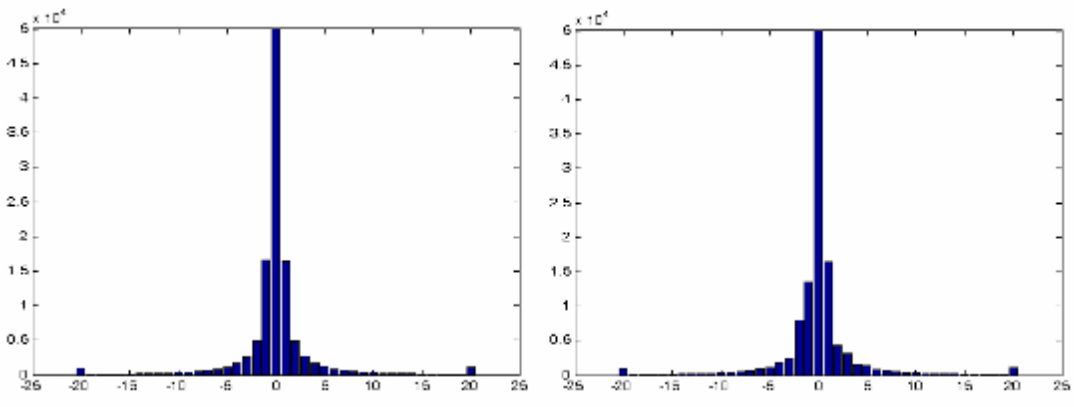
eğer katsayı 0 veya 1 değilse

gizli Mesajdaki sonraki LSB'yi al

katsayıdaki LSB'yi gizli Mesajdaki ile değiştir

Andres Westfeld ve Andreas Pfitzmann makalelerinde ki-kare saldırı tekniğini kullanarak J-Steg yazılımı ile yapılan gizlemeleri tespit edebilmişlerdir [32].

Şekil 4.10'da solda gizleme işlemi öncesi DCT katsayılarının histogramı verilmiştir. Bu histograma göre katsayıların miktarı simetriktir. J-Steg algoritmasında 1 ve 0 haricindeki katsayılar değiştirildiği için gizleme yapılan görüntünün histogramı alındığında normalde olması gereken simetri özelliğinin kaybolduğu gözlenmiştir. Gizleme yapıldıktan sonraki görüntünün histogramı Şekil 4.10'da sağdadır. Wayner, JPEG sıkıştırma tekniğinde normalde katsayıların çan eğrisi şeklinde olduğunu J-Steg ile gizleme yapılınca bu şeklin bozulduğunu söylemiştir [52].



Şekil 4.10. Veri gizlemeden önce ve sonra DCT katsayılarının histogramı [7]

Neils Provos OutGuess yazılımıyla J-Steg yazılımındaki ki-kare zafiyetini bir ölçüye kadar giderebilmiştir. Önerilen yöntemde DCT katsayılarını seçerken PRNG'den faydalanmıştır. Böylece histogramın simetrisini bir ölçüye kadar koruyabilmiştir. Kendisi ki-kare testinin gelişmiş bir versiyonu ile bu türlü gizlemeyi de tespit edebileceğini göstermiştir. Ayrıca araştırma kapsamında eBay ve usenet sitelerindeki yaklaşık 3 milyon görüntüde steganorafik içerik bulamamışlardır [9].

OutGuess Algoritması

Giriş: gizli Mesaj, Görüntü(JPEG)

Çıkış: stego Görüntü

paylaşılan gizli anahtar ile PRNG' yi başlat

gizli Mesajda saklanacak veri olduğu müddetçe

PRNG tarafından üretilen DCT katsayısını al

eğer katsayı 0 veya 1 değilse

gizli Mesajdaki sonraki LSB'yi al

katsayıdaki LSB'yi gizli Mesajdaki ile değiştir

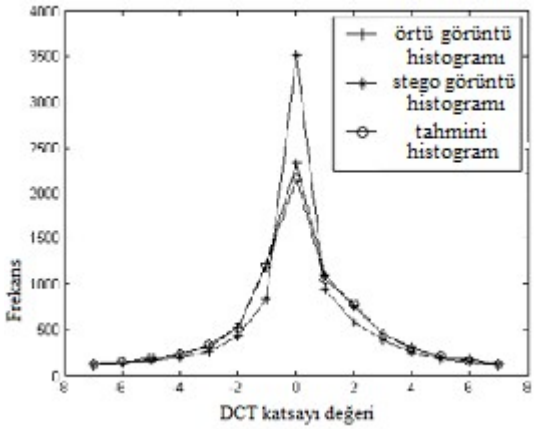
Andreas Westfeld görsel ve istatistiksel saldırılara karşı dayanıklı F5 algoritmasını geliştirmiştir [53]. Bu algoritma ile DCT katsayılarının histogramının bozulmaması sağlanmış ve veri DCT katsayılarına dağıtık bir şekilde saklanmıştır. Dağıtık saklamanın görüntü üzerinde örneklenmiş hali Şekil 4.11'de görülmektedir.



Şekil 4.11. Ardışık ve dağıtık saklama örnekleri [53]

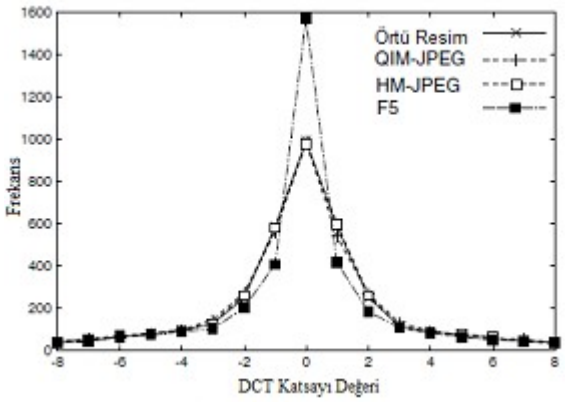
Bir yıl sonra Fridrich önerdiği steganaliz yöntemiyle F5 ile saklanan gizli verinin varlığını tespit etmiştir [54]. Bu yöntemde kilit nokta saklama yapılmadan önceki görüntünün histogramının tahmin edilmesidir. Bu da gizli bilgi içeren görüntüdeki

frekans alanındaki nicemlemeyi kaldırmak için her iki yönde 4 piksel kesip tekrar sıkıştırarak gerçekleştirilmektedir. Saldırı bu iki görüntünün histogramlarının karşılaştırılmasıyla olur. Bu karşılaştırmanın örneği Şekil 4.12'de görülmektedir.



Şekil 4.12. Histogram karşılaştırması [54]

Japonya'dan üç akademisyen, QIM (Quantization Index Modulation) ve HM (Histogram Matching) adında iki yeni yöntem önermişlerdir [55]. Bu yöntemler F5'e kıyasla daha fazla veri saklamaya imkan verirken F5'e saldırı sonucu ortaya çıkan histogram farkını ortadan kaldırmışlardır. Bu yöntemlerin histogramlarının F5 ile kıyaslaması Şekil 4.13'de görülmektedir.



Şekil 4.13. QIM, HM ve F5 histogram karşılaştırması [55]

Chang ve arkadaşları 2002 yılında yaptıkları çalışmalarında nicemleme matrisini

değiştirmişler ve saklama işlemini nicemlenmiş DCT katsayılarının orta frekansında yapmışlar. Jsteg algoritmasına kıyasla daha yüksek kapasite elde ettiklerini ve görüntünün kalitesinin kabul edilebilir seviyede olduğunu belirtmişlerdir [56].

Xiaoxia Li, Jianjun Wang 2007 yılında sürü zekasını temel alan evrimsel bir hesaplama modeli olan parçacık sürü optimizasyonu yöntemini ideal bir değişiklik matrisi türetmek amacıyla kullanan yöntemlerini önermişlerdir. Bu yöntemle Chang'ın yönteminden daha yüksek saklama kapasitesi ve daha kaliteli görüntü elde etmişlerdir [57].

İran'lı üç akademisyen 2009 yılında yayınlanan çalışmalarında ki-kare saldırısını aldatabilen bir yöntem önermişlerdir [58]. Bu yöntem ardışık ikili DCT katsayı değerlerinin gruplanmasına (2-3, 4-5 gibi) dayanmaktadır. Bir gruba ait iki sayı bulunduğunda bunlara bir bit saklama yapılmaktadır. Bu da saklama kapasitesini yarıya düşürmektedir. Saklama sayılar grubun dışına çıkmayacak şekilde yapılmaktadır böylece bir gruptaki değişiklik diğerini etkilememektedir. Örneğin saklamadan önce 3-2 değerine sahip ikilinin saklamadan sonra alabileceği değerler 2-2, 2-3, 3-2, 3-3 olabilir. Her grubun katsayıları ikişer ikişer incelenirse ve bu ikililer ki-kare saldırısı hesaba katılarak değiştirilirse orijinal görüntü ile gizleme yapılmış görüntünün histogramları arasında çok ufak bir fark olabilir.

Amerika Birleşik Devletlerinden dört akademisyen yaptıkları çalışmada mevcut istatistiksel steganaliz yöntemlerinin tespit edemediği yöntemlerini sunmuşlardır [59]. Bu yöntem örtü görüntünün istatistiklerinin korunacağı bir şekilde yüksek karmaşıklığa sahip DCT bloklarının ve katsayılarının seçilmesine dayanmaktadır.

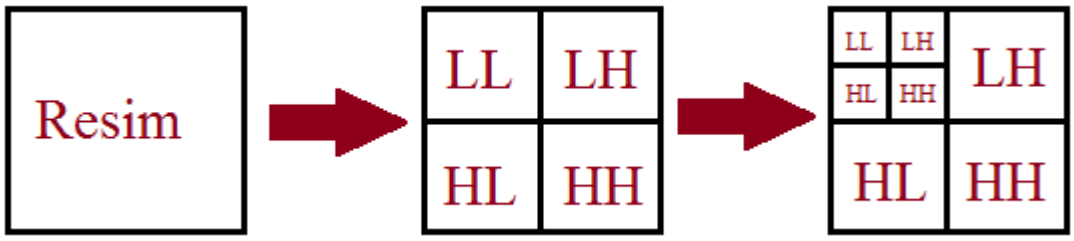
DWT Steganografi

Görüntü dosyalarında DWT steganografi JPEG 2000 dosya türü üzerinde uygulanmaktadır. DWT'nin DCT'ye göre avantajı yüksek oranlı sıkıştırmalarda daha kaliteli görüntü üretebilmesidir. DWT'nin insan görme sistemiyle daha uyumlu olması bu alanda yapılan çalışmaların artmasına sebep olmaktadır. DCT piksel

blokları üzerinde çalışır ve bu saklama işlemindeki ufak bir hata blok geçişlerinde istenmeyen sonuçlar doğurabilir. DWT ise piksel blokları yerine görüntüyü frekans bantlarına bölmektedir. Bu da daha etkin sıkıştırma sağlamaktadır. Bu frekans bantları şunlardır:

LL - *Yatay ve dikey alçak geçiş*
LH - *Yatay alçak geçiş ve dikey yüksek geçiş*
HL - *Yatay yüksek geçiş ve dikey alçak geçiş*
HH - *Yatay ve dikey yüksek geçiş*

Bu bantlardan HL, LH, HH sırasıyla görüntünün yatay, dikey ve çapraz özelliklerini temsil etmektedir. İnsan gözü düşük frekanslara daha duyarlı olduğu için LL bandı dışındaki bantlar veri saklama işlemi için kullanılabilir. Şekil 4.14 görüntünün bantlara dönüştürülmesi işlemini göstermektedir.



Şekil 4.14. Ayrık dalgacık dönüşümü bantların gösterimi

Ahmed A. Abdelwahab ve Lobna A. Hassaan DWT steganografide yeni bir teknik önermişlerdir. Bu teknikte örtü görüntü ve gizli görüntü DWT ile 4 banda bölünür. Bu bantlar çakışmayacak bloklara ayrılır. Örtü görüntü blokları gizli görüntü bloklarından çıkarılır. Bu bloklara hata blokları denilir. Bu bloklar daha sonra yatay banttaki en iyi eşleşen blokla yer değiştirilir [60].

Bir diğer çalışmada Abdelwahab'ın yatay banda uyguladığı yöntem diğer bantlar üzerinde denenmiş ve çapraz bandın diğer bantlardan daha iyi PSNR (Peak Signal to Noise Ratio) değeri verdiği görülmüştür [61].

Bir çalışmada yüksek kapasiteye sahip, görüntüyü orijinal haline geri çevirebilen bir yöntem önerilmiştir [62]. Görüntü bloklara ayrılmış ve bloklar üzerindeki yüksek frekansa sahip DWT katsayıları üzerinde adaptif bir yöntem uygulanmıştır. Saklanan bilgiye görüntüdeki yan etki de eklenmiş ve böylece orijinal görüntünün elde edilmesi sağlanmıştır.

Amirkabir üniversitesinden üç akademisyen 2011 yılında yayınlanan çalışmalarında DWT üzerinde çalışan güçlü adaptif bir yöntem önermişlerdir [63]. Bu yöntemde görüntünün LH, HL ve HH bantlarını 8x8'lik bloklara ayrılır ve her blokun 8 bit için bit düzlemi çıkarılır. Bu düzlemlere göre karmaşıklık hesabı yapılır. Karmaşıklık hesabı sonucunda blokların saklama kapasiteleri bulunur. Saklama kapasitesi birden büyük olanlarda yuvarlama yöntemi, bir olanlarda ise üçte bir yöntemi kullanılır. Eğer saklama sonucunda oluşan blok ile orijinal blok arasında belirlenen eşik değerinden düşük miktarda fark varsa saklama yapılır. Saklanan bir adedi bilgisi de ilgili bloktaki ilk pikselin son üç bitine gömülür. Bu yöntemin aynı kapasiteyi sağlayan yöntemlere göre saldırılara karşı daha dayanıklı olduğu tespit edilmiştir.

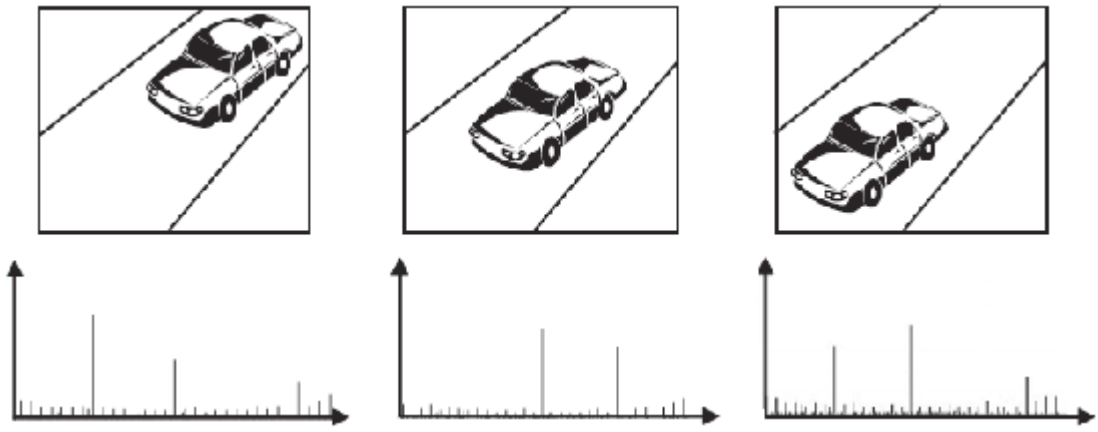
4.4. Video Steganografi

Ses dosyalarının ve hareketsiz görüntü dosyalarının kapasiteleri kısıtlıdır. Bu kapasite kısıtı saklanabilecek bilgi miktarını da kısıtlamaktadır. Dosya sistemindeki bir klasörü saklamak isteyen kişi çoğu klasör için bunu gerçekleştiremeyecektir. Bunun sebebi saklanmak istenen klasörün genelde örtü görüntünün saklama kapasitesini aşmasıdır. Bu kapasite kısıtını aşmak için video dosyaları kullanılmalıdır.

Video dosyaları, görüntü dosyalarına benzemektedir ancak aralarında önemli farklar da vardır. Video dosyalarında daha çok piksel vardır. Bu da videoların saklama açısından kapasitesini arttırmaktadır. Bunun dışında videolarda genelde görüntünün ufak farklarla değişmesi tekrar eden gereksiz bilgiye sebep olmaktadır. Bu gereksiz bilgiyi önlemek için kullanılan sıkıştırma yöntemleri saklama işlemini zorlaştırmaktadır.

4.4.1. AVI dosyalarında steganografi

Kuala Lumpur üniversitesinden dört araştırmacı yaptıkları çalışmada AVI video formatını kullanarak yüksek oranlı veri saklama gerçekleştirmişlerdir. Kullandıkları yöntem insan gözünün maviye daha hassas olmasından faydalanmakta ve bir pikselin kırmızı ve yeşil alanına 3 bit mavi alanına 2 bit bilgi saklayarak 1/3 oranında saklama yapmayı başarmışlardır. Ayrıca insan gözünün deniz veya uzay gibi geniş ve düz alanlardaki değişimlere pek çok nesnenin bulunduğu karışık alanlardaki değişimlerden daha hassas olduğu için kalabalık videoların saklama için daha uygun olduğunu söylemişlerdir [64].



Şekil 4.15. Ardışık üç çerçeve ve renk bileşen histogramları [16]

Özdemir Çetin ve Turan Özcerit 2009 yılındaki araştırmalarında saklama yapılacak pikseli seçerken çerçevelerin renk histogramlarını kullanan yeni bir yöntem önermişlerdir [16]. Bu yönteme göre çerçevelerin renk bileşenlerinin histogramı hesaplanır ve ardışık çerçeveler arasında bu histogram değerleri karşılaştırılır (Şekil 4.15) ve bir eşik değeri elde edilir. Bu eşik değeri piksel seçiminde kullanılır. Bu yöntemi çerçeve tabanlı ve blok tabanlı olmak üzere uygulamışlardır. Blok tabanlı yöntem algılanabilirlik açısından daha başarılı sonuç vermiştir.

4.4.2. MPEG dosyalarında steganografi

Bir çalışmada sadece I tipi çerçevelerden faydalanılmış ve DCT işleminden sonra belirli bir eşik değerinin üzerindeki katsayılara saklama önerilmiştir [22]. P ve B çerçevelerinin kullanılmaması bu çalışmadaki eksikliklerdendir.

Çin Halk Cumhuriyetinden üç araştırmacı önerdikleri algoritmada kontrol verisi olarak adlandırdıkları GOP'un saklama kapasitesini ve gizli bilginin başlangıç noktasını tutan bilgileri I çerçevesine saklamışlardır. Gizli bilgiler ise P ve B tipi çerçevelere saklanmıştır. Hareket vektöründeki değişikliğin belirli bir eşik değerinin üstünde olması şartıyla vektörün son biti saklama amacıyla kullanılmıştır. Sonuç olarak 352 x 240 boyutunda bir videoda bir GOP için 0.5 KB'lik veriyi 40 dB'nin altında gürültüyle saklamayı başarmışlardır [65].

Hussein A. Aly 2011 yayınlanan makalesinde steganalize karşı savunmasız olduğundan ötürü I tipi çerçeveleri kullanmamış sadece P ve B tipi çerçevelere saklama yapmıştır. Diğer çalışmalardan farklı olarak, saklama yapılacak vektörlerin seçimini yaparken vektörün büyüklük veya açısal özelliklerini değil vektörle ilişkili makro blok tahmin hata miktarını kullanmıştır [66].

Sherly A. P. ve Amritha P. P. 2010 yılında yayınlanan çalışmalarında MPEG videolarında sıkıştırma alanında saklama yapan yeni bir şema önermişlerdir. Veri saklama için üç yönlü piksel değer değişimi algoritmasını kullanmışlardır. Sonuç olarak yüksek kapasite ve insan gözüyle algılanamamasını sağlamışlardır. 288 x 352 piksellik orana sahip bir videoda I tipi çerçevelere yaklaşık 6 KB, B tipi çerçevelere yaklaşık 1,5 KB, P tipi çerçevelere ise yaklaşık 3KB veri saklamayı başarmışlardır. Elde edilen gürültü miktarı ise 60 dB civarındadır [67].

4.4.3. Mevcut video steganografik yazılımlar

Günümüzde internette yüzleri aşan sayıda steganografik yazılım geliştirilmiştir. Bunların çoğunluğu görüntü ve ses steganografisi üzerinedir. Bu uygulamaların

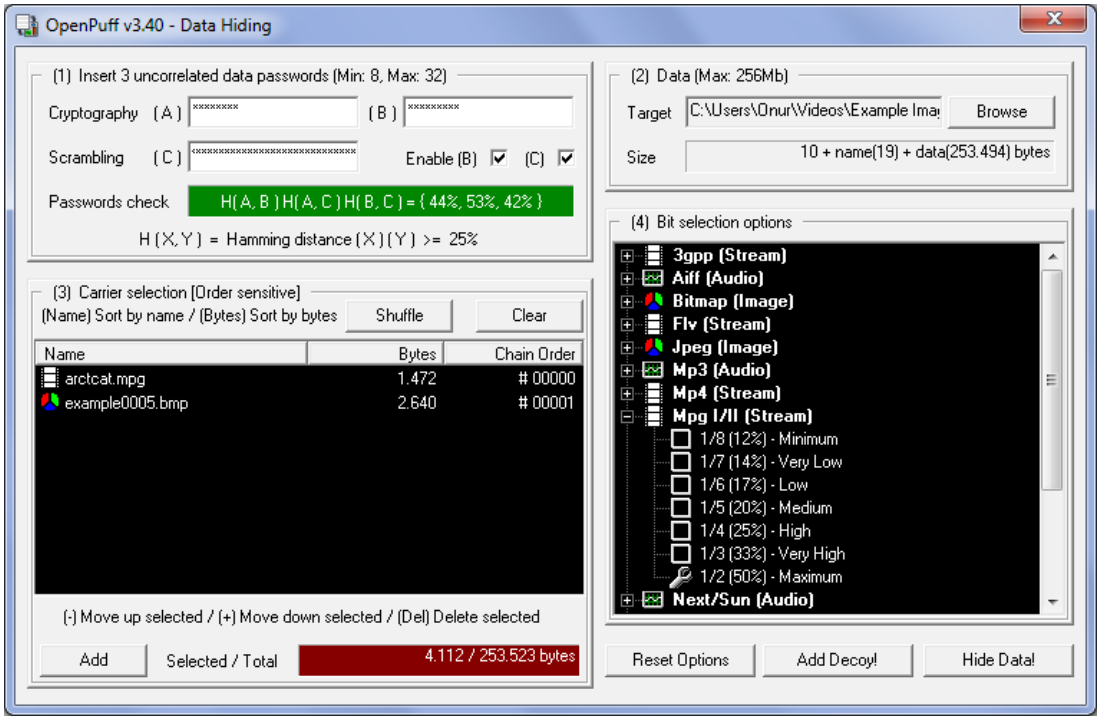
büyük bolümünü barındıran bir web sayfası da mevcuttur [68]. Video dosyalarına saklama yapabilen yazılım sayısı oldukça azdır. Bu çalışma yapıldığı zamanda iki tane bulunmuştur. Bunlar OpenPuff [69] ve MSU Stego'dur [70]. İkisinden başarı olarak öne çıkan OpenPuff yazılımıdır.

OpenPuff

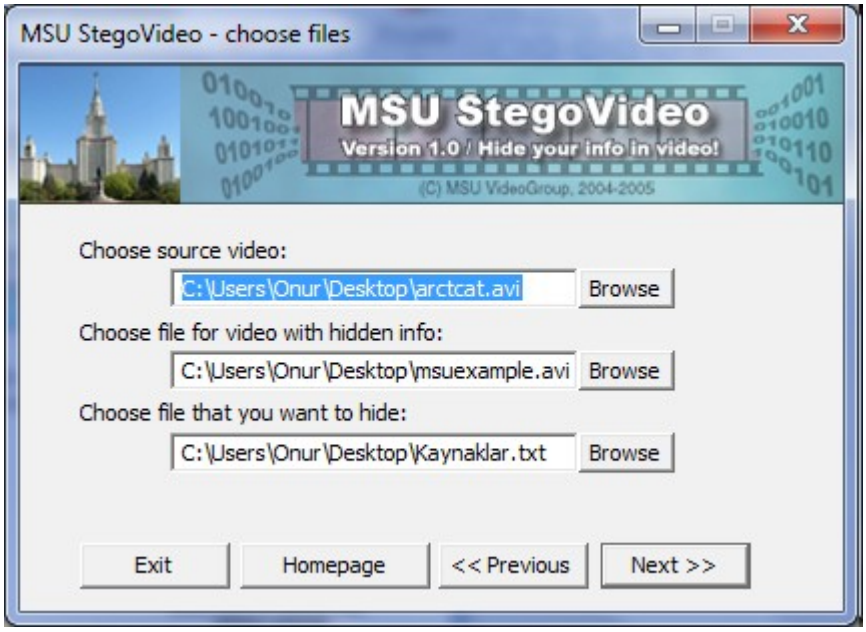
İtalyan Cosimo Oliboni tarafından C++ programlama dili kullanılarak geliştirilen bu araç pek çok formatı desteklemektedir. Saklama yapmadan önce şifrelemeyi zorunlu kılmıştır. Sadece dosya saklamaya izin vermektedir. Klasör olarak saklama yaptırmamaktadır. Taşıyıcı ortam olarak birden fazla dosya seçilmesine izin vermektedir. Bu durumda dosyaların sırasını hatırlamak gerekmektedir. Saklama oranının seçilmesine de izin vermektedir. Ancak video dosyalarında saklama kapasitesi çok düşük çıkmaktadır. Örneğin 7.5 MB boyutunda bir MPEG dosyasına en yüksek saklama oranı seçildiğinde 1472 bayt saklama yapmaktadır. Güvenliğin üst düzeyde olması için şifreleme kullanılmıştır ve gürültü eklenmiştir. Ortam bağımsız, yüksek güvenlik sağlayan güncel bir yazılımdır.

MSU StegoVideo

Sadece AVI dosyalarına saklama yapabilmekte fakat tüm AVI dosyalarını açamamaktadır. Saklama kapasitesi oldukça azdır. Örneğin 340 Mbayt'lık bir AVI 1.0 dosyasına 6689 bayt'lık veri saklayabilmektedir. Saklanacak dosya olarak metin dosyası önerilmektedir. Kayıplı saklama yapabilmektedir. Çıkarma işlemi sonucunda elde edilen metindeki harflerin bir kısmı bozulmaktadır. Ayrıca saklama işlemi sonucunda videodaki ses kaybolmaktadır. Öne çıkan özelliği farklı kodekler kullanılarak yapılan sıkıştırma işlemine dayanıklılık gösterebilmesidir. Videoda tekrar eden veri miktarı arttıkça saklanan bilgiyi çıkarma işlemindeki hata oranı azalmaktadır.



Şekil 4.16. OpenPuff yazılımı arayüzü



Şekil 4.17. MSU StegoVideo yazılımı arayüzü

Yukarıdaki yazılımların ve bu çalışma kapsamında gerçekleştirilen uygulamanın özelliklerinin karşılaştırması Çizelge 4.4'de gösterilmiştir.

Çizelge 4.4. MSU StegoVideo, OpenPuff ve StegoFolderLock'un karşılaştırması

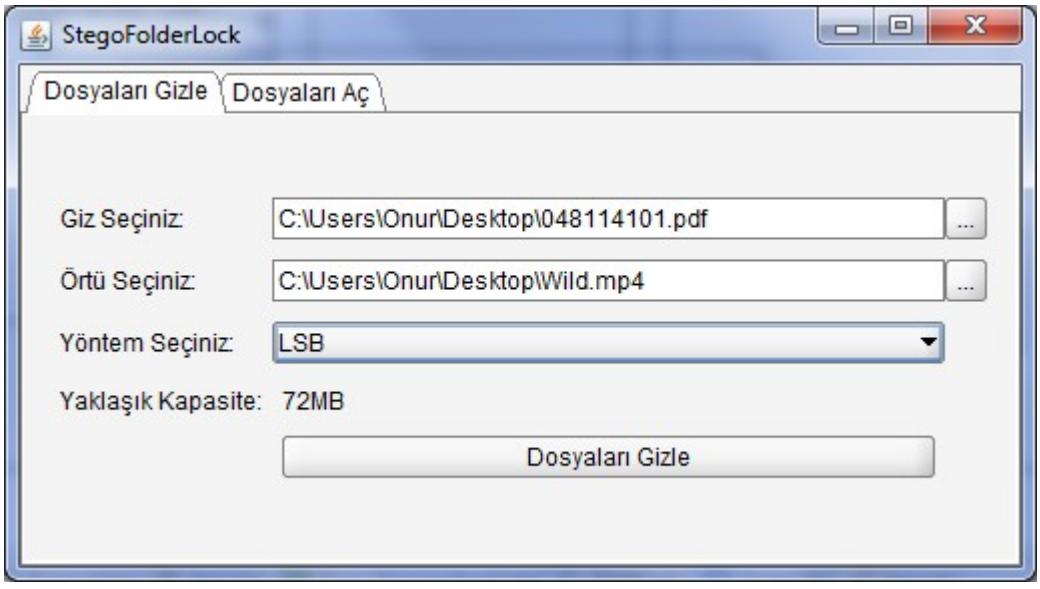
Yazılım	Ortam	Yöntem	Saklanan nesne	Çoklu saklama	Kapasite	Webcam ile saklama
MSU StegoVideo	avi	Bilinmiyor	Metin dosyaları	yok	Video boyutuna bağlı	yok
OpenPuff	bmp,jpg,pcx,png,tga,aiff, mp3,next/sun,wav,3gp, mp4,mpg,vob,flv,pdf,swf	Adaptif LSB, bit seçme	Tüm dosya türleri ve klasörler	var	256 Mbayta kadar	yok
StegoFolderLock	bmp, wav, avi	LSB, LSBM, LSD	Tüm dosya türleri ve klasörler	yok	Teorik olarak sınırsız	var

5. GELİŞTİRİLEN UYGULAMA (StegoFolderLock)

Bu yazılımla hedeflenen steganografik uygulamalar için esnek bir kütüphane oluşturulması ve bu kütüphanenin doğrudan kullanımı için bir de arayüz sunulmasıdır.

Kütüphane LSB, LSBM ve LSD yöntemlerini desteklemektedir. Bu yöntemler Bölüm 3.3.1'de anlatılmıştır. Kütüphane WAV, BMP ve AVI sıkıştırma kullanmayan ham medya ortamlarını desteklemektedir. Bu ortamlar hakkında Bölüm 2'de bilgi verilmiştir. Kütüphanenin esnek tasarımı sayesinde yeni yöntem ve ortamları entegre etmek kolaydır. Aynı zamanda uygulama web kamerasından kayıt yaparak anlık olarak bilgi saklama kapasitesine de sahiptir.

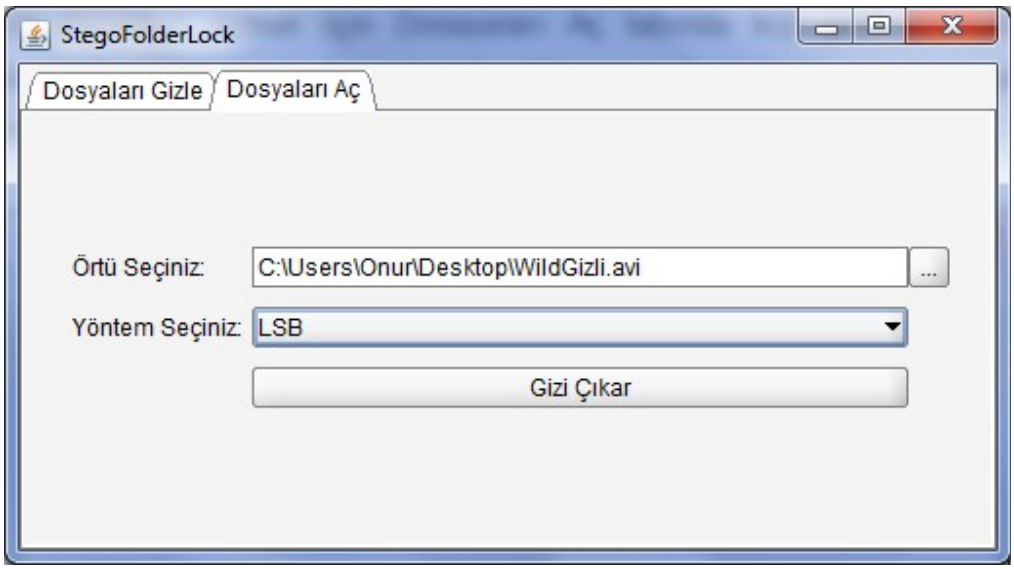
Uygulamanın arayüzü aşağıdaki gibidir. Bu ekranda kişi bilgisayarındaki gizlemek istediği herhangi bir dosya veya klasörü seçer. Sonra gizli dosyaların saklanacağı ortamı (görüntü, video veya ses dosyası) ve saklama işleminde kullanılacak yöntemi seçer. Seçimi tamamladığında seçilen ortama seçilen yöntemle yaklaşık ne kadar bilgi saklanabileceğini yazılım gösterecektir. Bu kapasite aşılırsa işlem başarısız olur. Dosyaları gizle düğmesine bastığında uygulama gizleme işlemini yapar (Şekil 5.1).



Şekil 5.1. Stegofolderlock dosya gizleme arayüzü

Web kamerasından kayıt için ortamın seçildiği "Örtü Seçiniz" metin alanına canlı yazılması yeterlidir. Böylece yazılım saklanacak bilgi için gereken miktarda web kamerasından kayıt yapacak ve örtü videoyu oluşturacaktır.

Gizli dosyaları açmak için Dosyaları Aç tabında kişi gizli dosyaların bulunduğu ortamı seçer (Şekil 5.2). Gizi Çıkar düğmesine bastığında uygulama gizli dosyaları ilgili adrese çıkarır.



Şekil 5.2. Stegofolderlock gizli dosyaları çıkarma arayüzü

5.1. Kütüphanede Kullanılan Teknolojiler

Bu uygulama ile hedeflenen ortam ve yöntem bağımsız bir kütüphane oluşturmaktır. Ortam bağımsızdan kasıt kütüphane görüntü, ses ve video dosyalarına veri saklama ve çıkarma yapabilmelidir. Yöntem bağımsızdan kasıt ise gizleme yapılırken kullanılan algoritma değiştirilebilir olmalıdır.

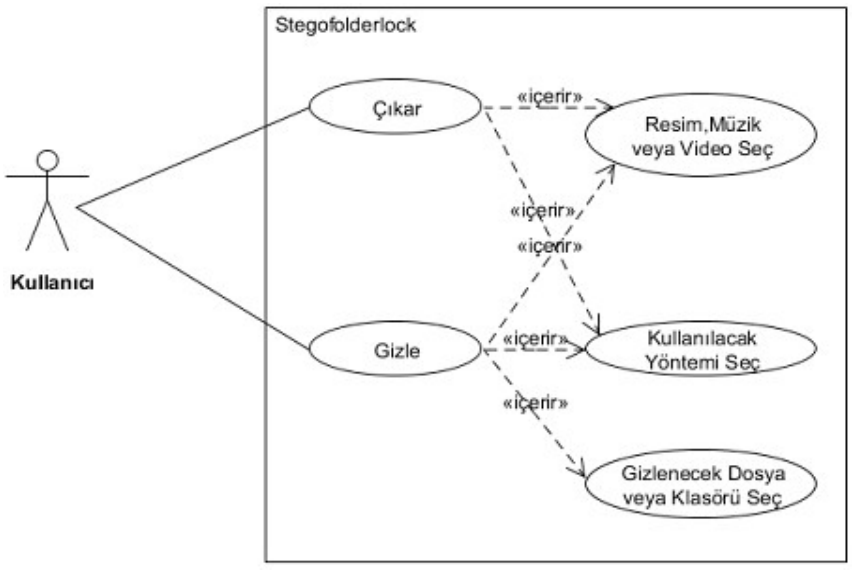
Uygulama Java programlama diliyle gerçekleştirilip platform bağımsız olması sağlanmıştır. Uygulama ortam ve yöntem bağımsız olacağı için esnek bir tasarıma gereksinim vardır. Bu sebeple çeşitli tasarım desenleri kullanılmıştır. Eclipse Indigo entegre geliştirme ortamı kullanılmıştır. Mylyn, Eclipse eklentisi görev yönetimi

amacıyla kullanılmıştır. Kaynak kod yönetimi Subversion ile gerçekleştirilmiştir. UML diyagramlarının çiziminde Umlet, Eclipse eklentisi olarak kurulup kullanılmıştır. Video çatısı (framework) olarak Xuggler tercih edilmiştir.

Video işleme amacıyla java ortamında kullanılabilecek başlıca çatılar (framework) JMF (Java Media Framework), FMJ (Freedom for Media in Java) ve Xuggler'dır. JMF, Sun tarafından geliştirilen ve 2000'li yıllardan beri pek güncellenmeyen kodek desteği kısıtlı bir çatıdır. FMJ ise güncel fakat dokümantasyonu yetersiz bir seçenek olarak ortaya çıkmaktadır. Xuggler geniş kodek desteği, detaylı dokümantasyonu ve öğretici videoları ile hızla öğrenilebilecek ve uygulanabilecek seçenek olarak öne çıkmaktadır.

5.2. Kütüphanenin Tasarımı ve Çalışma Mantığı

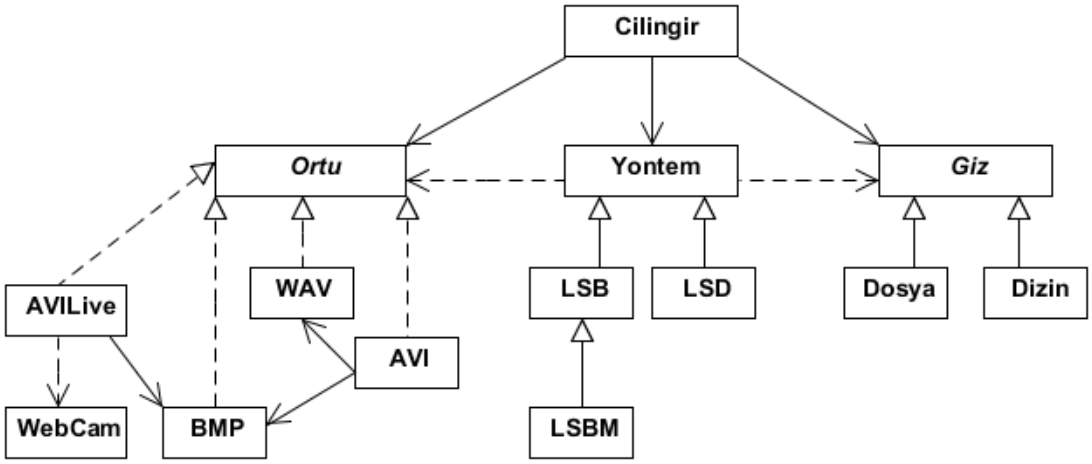
Kullanıcının gereksinimleri seçilen dosya veya klasörün, seçilen yöntem kullanılarak, seçilen görüntü, ses veya video dosyasına saklanması ve daha sonra saklama yapılmış ortamdan saklamada kullanılan yöntem seçilerek gizli bilgilerin çıkartılmasıdır. Bu gereksinimleri içeren kullanım durumu diyagramı Şekil 5.3'de görülmektedir.



Şekil 5.3. Gereksinimler kullanım durumu diyagramı

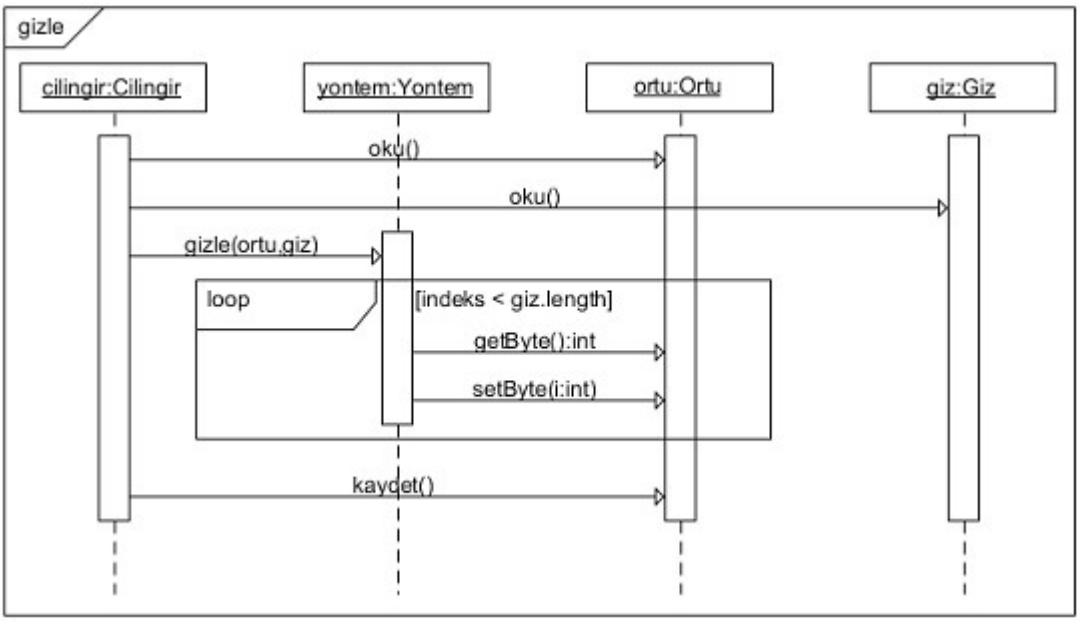
Yazılımda saklama yapılacak ortam değişebilmeli ve saklama yöntemi de değişken olmalıdır. Bunlardan ötürü Bridge tasarım deseni kullanılmıştır. Bu şablon kullanımı, soyut sınıflar ve bunların gerçekleştirimleri birbirinden ayrılmak isteniyorsa tavsiye edilir. Bridge tasarımı esnekleştirmektedir. Şekil 5.4.'de kullanılan sınıflar ve genel tasarımı içeren sınıf diyagramı verilmiştir.

Sistemde saklama ve çıkarma işlemlerinin yönetimi *Cilingir* sınıfı tarafından yapılmaktadır. Şekil 5.4'de görüldüğü gibi *Cilingir* sınıfı *Ortu*, *Yontem* ve *Giz* sınıflarına referans tutmaktadır ve bunları gizleme ve çıkarma işlemleri için yönetmektedir. Sistemin karmaşıklının istemcilerden gizlenmesini sağlar. Facade tasarım desenine örnektir.



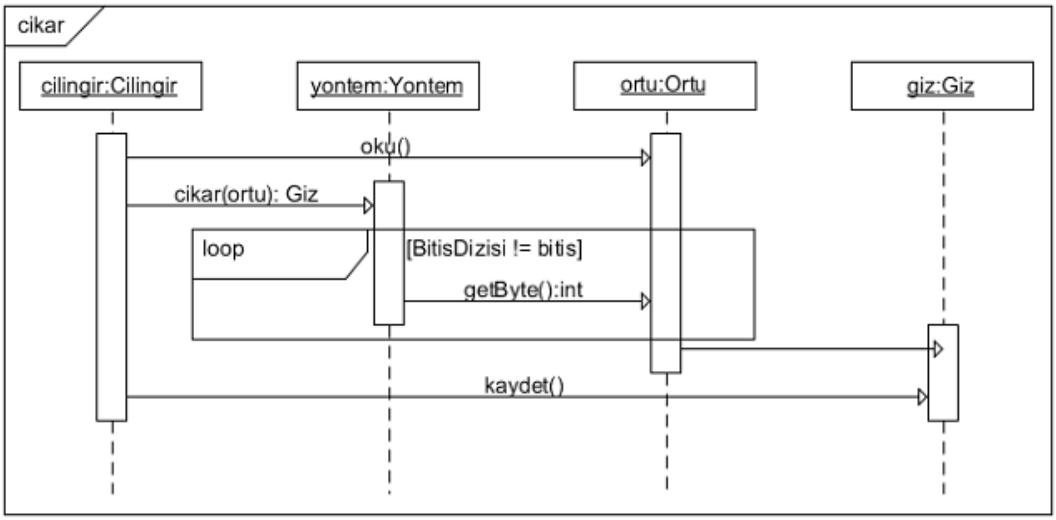
Şekil 5.4. Sistem genel sınıf diyagramı

Gizleme işlemi yapılırken *giz* ve *ortu* nesneleri okunarak hafızaya alınır ve *yontem* nesnesi üzerindeki *gizle(ortu:Ortu, giz:Giz)* metodu okunan nesneler parametre olarak geçilerek çağırılır. *Yontem* nesnesi *ortu* nesnesinden *getbayt():int* metodu ile aldığı bayt'a *giz* nesnesinden aldığı gizli bilginin bir kısmını saklar ve saklı bilginin bulunduğu bayt'ı yazması için *ortu* nesnesine *setbayt(i:int)* metodu aracılığı ile gönderir. Bu işlem *giz* nesnesinde gizlenecek bilgi kalmayıncaya kadar devam eder. Son olarak gizli bilgiyi içeren *ortu* nesnesi dosya sistemine kaydedilir.



Şekil 5.5. Gizleme işlemi ardışıl diyagramı

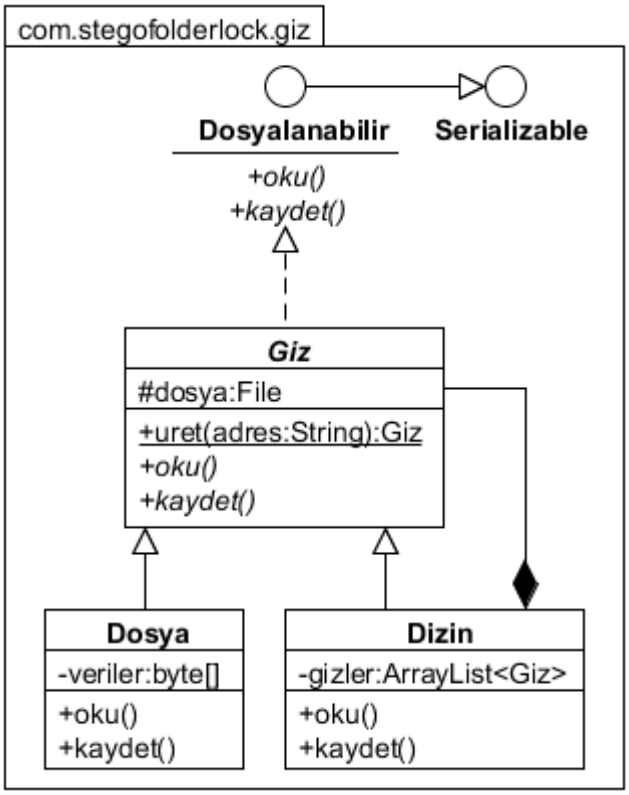
Çıkarma işlemi yapılırken *ortu* nesnesi okunarak hafızaya alınır ve *yontem* nesnesi üzerindeki *cikar(ortu:Ortu):Giz* metodu okunan nesne parametre olarak geçilerek çağrılır. *Yontem* nesnesi *ortu* nesnesinden *getbyte():int* metodu ile aldığı bayt'daki gizli bilgiyi alarak hafızada *giz* dizisini oluşturmaya başlar. Bu işlem çıkarılmamış gizli bilgi kalmayıncaya kadar devam eder. Son olarak *giz* dizisi *giz* nesnesine dönüştürülür ve *kaydet()* metoduyla dosya sisteminde oluşturulur.



Şekil 5.6. Çıkarma işlemi ardışıl diyagramı

Kütüphanede dosya sistemindeki herhangi bir dosya veya klasör *giz* olarak kabul edilmiştir. Bunlar yazılım tarafından gizlenebilmelidir. Saklanan nesne *giz* nesnesidir. *Giz* soyut sınıfı *oku()* ve *kaydet()* metotlarını barındıran *Dosyalanabilir* arayüzünü gerçekleştirmiştir. *Giz* soyut sınıfı içerisindeki *uret(adres:String)* metodu *giz* nesnesi üretme işlemiyle görevlendirilmiştir. Bu metot Factory Method tasarım deseni kullanılarak gerçekleştirilmiştir.

Dizin ve *Dosya* sınıfları dosya sistemindeki dosya ve dizine karşılık gelmektedirler. *Giz* soyut sınıfından türerler. *Dizin* birden fazla *giz* nesnesi içerebilir. Bu yapıyı kurmak için Composite tasarım deseni kullanılmıştır. Şekil 5.7.'de bu tasarımın sınıf diyagramı görülmektedir.

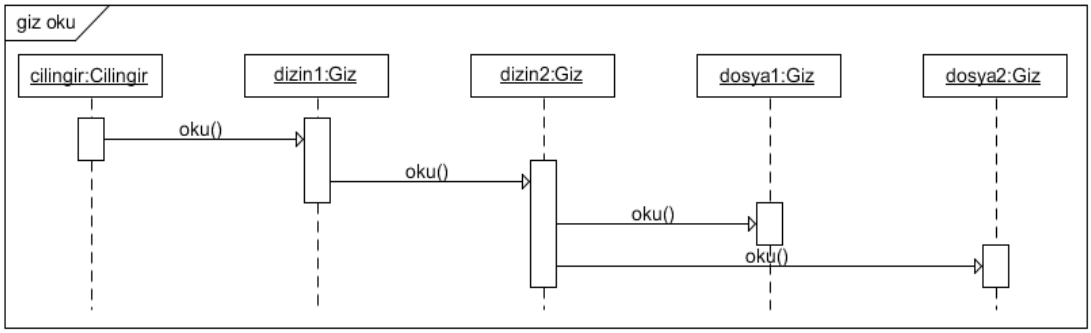


Şekil 5.7. Giz paketinin sınıf diyagramı

Giz soyut sınıfında *uret(adres:String):Giz*, *oku()*, *kaydet()* olmak üzere üç metot vardır. *Dosyalanabilir* arayüzü, dosya sistemindeki birimleri temsil ettiği için her türevinin bir adresi olacaktır. Bu adres *giz* nesnelerinde, *file* nesnesi içerisinde

tutulur. *Giz* nesnesinde *uret(adres:String)* metodu çağrıldığında kök nesnenin adresi hafızaya alınır. Böylece tüm verilerin hafızaya alınması okuma anına ertelenmiş olur.

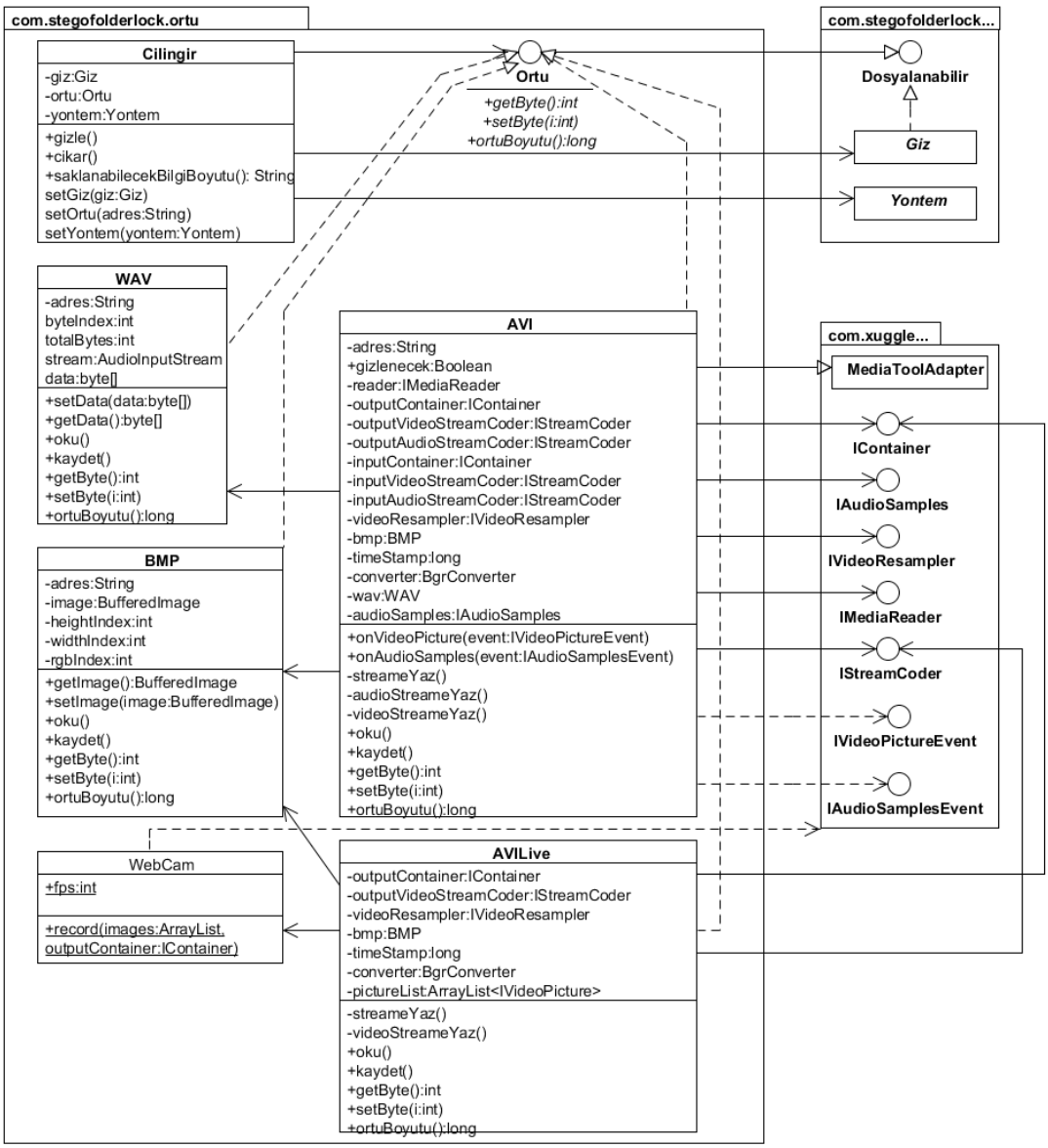
Cilingir nesnesi gizleme işlemine başlamadan önce gizlenecek nesneleri hafızaya almalıdır. Bu işlem *giz* nesnesinde oku() metodunun çağrılmasıyla ve bu kompozit nesnenin barındırdığı diğer *giz* nesnelerine iç içe *oku()* çağrılarının yapılmasıyla gerçekleştirilir. Giz okuma işleminin ardışıl diyagramı Şekil 5.8'deki gibidir.



Şekil 5.8. Giz okuma işlemi ardışıl diyagramı

Gizleme yapılacak ortamlar *ortu* paketinde tutulmuştur. *BMP, WAV, AVI, AVILive* gibi sınıflar *Ortu* arayüzünü gerçekleştirmelidir. *Ortu* arayüzü saklama yapılacak sıradaki baytı vermekle ve almakla yükümlüdür. *Yontem* soyut sınıfından türeyen bir sınıf bu baytı alır ve gerekli değişikliği yapar ve geri verir. Bütün veri gizlendiğinde *Ortu* arayüzünün türevindeki *kaydet()* metodu çağrılır ve gizli bilgiyi içeren ortam kaydedilir.

AVI sınıfı oluşturulurken xuggle video kütüphanesinden faydalanılmıştır. Video dosyasının içerisindeki ses ve görüntü paketleri dinleyiciler ile yakalanır ve hafızaya alınır. Görüntü paketi *BMP* sınıfına delege ettirilir. *BMP* nesnesinde saklama tamamladığında bu bit eşlem içeriği gizli video dosyasına paket olarak yazdırılır ve bir sonraki paket alınır. Ses paketi *WAV* sınıfına delege ettirilir. *WAV* nesnesinde saklama tamamladığında bu ses içeriği gizli video dosyasına paket olarak yazdırılır ve bir sonraki paket alınır. Bu işlemler gizlenecek veriler bitene kadar devam eder. *AVI* sınıfının kullandığı sınıflar Şekil 5.9'da görülmektedir.



Şekil 5.9. Ortu paketi sınıf diyagramı

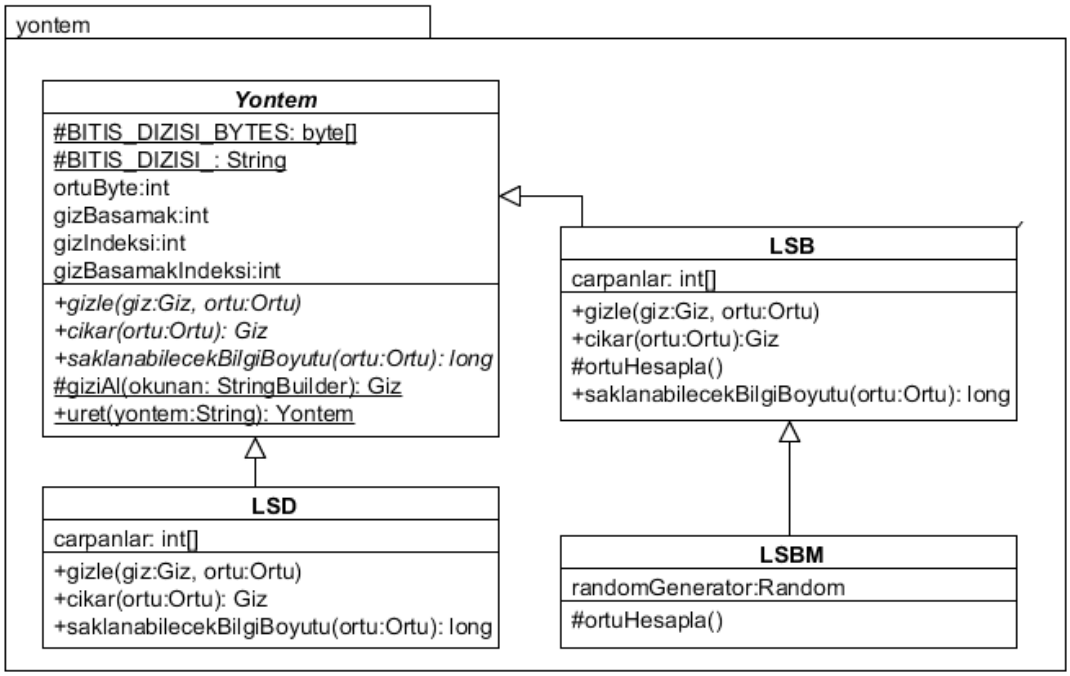
AVILive sınıfı web kamerasından alınan videoya gerçek zamanlı olarak saklama yapmayı sağlar. Bu yöntemin bir avantajı kayıt esnasında anlık olarak saklama yapılması sayesinde videonun aslına erişimin engellenmesidir. Böylece orijinali video ile karşılaştırılarak steganaliz uygulanması engellenmiş olur. Bir diğer avantajı ise saklanacak verinin boyutuna göre videoyu üretmesidir. Bu sayede optimum örtü boyutu sağlanmış olur.

AVILive sınıfı multi-thread bir yapıya sahiptir. Bir thread *WebCam* sınıfı ile çekilen

görüntüleri bir listeye atarken diğer thread listeye gelen paketlerde saklama yaparak çıktı videoya yazmaktadır. Bu işlemler saklama tamamlanana kadar devam eder.

Ortu üretiminden *Cilingir* sınıfı sorumludur. Bu sınıfta da Factory Method tasarım şablonu kullanılmıştır.

Saklama yöntemleri yontem paketinde toplanmıştır. *LSB, LSD* gibi saklama işlemi yapan sınıflar *Yontem* soyut sınıfından türemelidir. Ana olarak saklama ve çıkarma işlemini yapan metotları ihtiva ederler. Yontem soyut sınıfı arayüzü *gizle(ortu:Ortu,giz:Giz), saklanabilecekBilgiBoyutu(ortu:Ortu):long ve cikar(ortu:Ortu):Giz* metotlarına sahiptir. Bu sınıf saklama işlemini yapan sınıf olduğu için gizli nesneye ve örtü nesneye sahip olmalıdır. Sınıf diyagramı Şekil 5.10'da görülmektedir.



Şekil 5.10. Yontem paketi sınıf diyagramı

Örneğin *LSB* sınıfı gizleme yaparken *ortu* nesneden *getbyte() metodu* ile bayt alır. *Giz* nesnesinden ise bir bit alır. Bu biti *ortu* nesnesinin bayt'ının son biti ile değiştirir ve *setbyte(int gizlibayt) metodu* ile bayt'ın son halini yazdırır. Gizli bilginin tamamı

saklandıktan sonra ise bilginin bittiğini gösteren, denk gelmesi zor bir bayt deseni saklanır. Bu şekilde bilginin bittiğini anlarız. Saklama işlemi bittiğinde *ortu* nesne kaydedilir.

LSBM yöntemi LSB yönteminden türemektedir. *ortuHesapla()* şablon metodunu değiştirerek kullanmaktadır. LSBM yöntemi sahte rastgele sayı üreteci kullanmamaktadır.

Gizli bilginin çıkarılması işlemi yapılırken *ortu* nesneden bayt'lar alınır ve gizli bitler çıkarılır. Bitiş deseni geldiğinde hafızadaki bayt'lar *giz* nesnesine çevrilir. Son adımda *giz* nesnesinin *olustur() metodu* kullanılarak gizli nesne dosya sisteminde oluşturulur.

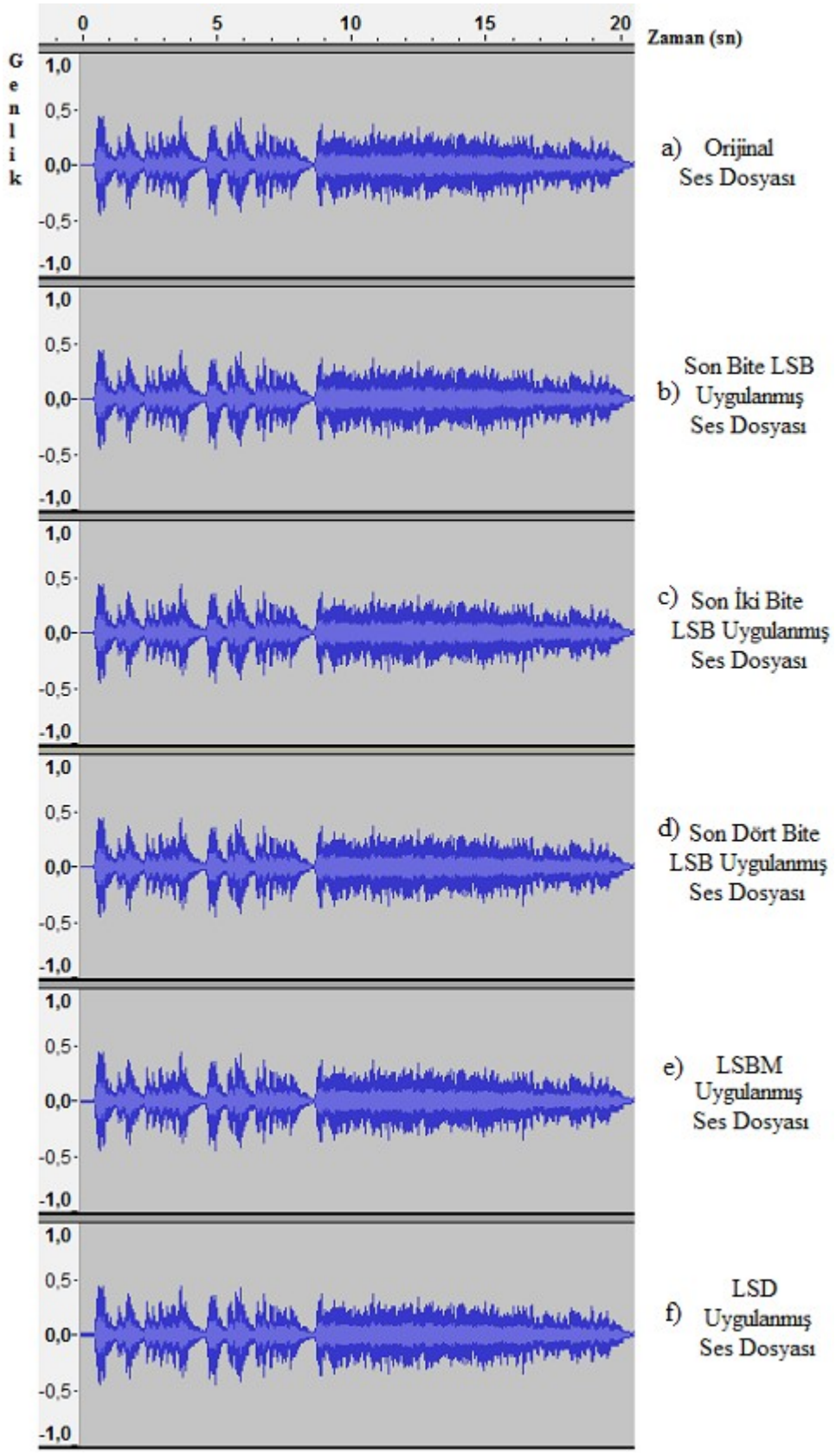
6. DENEYLER

Stegofolderlock uygulaması ile wav, bmp ve avi dosya formatlarına veri gizleme yapılabilmektedir. Bu uygulama aracılığı ile gerçekleştirilmiş gizleme işlemleri sonucunda oluşan dosyalar üzerinde yapılan deneyler bu bölümde verilmiştir. Görüntü dosyaları üzerinde yapılan deneylerde kullanılan saldırı yöntemleri bölüm 4.3.1'de LSB yöntemine saldırılar başlığı altında tanıtılmıştır.

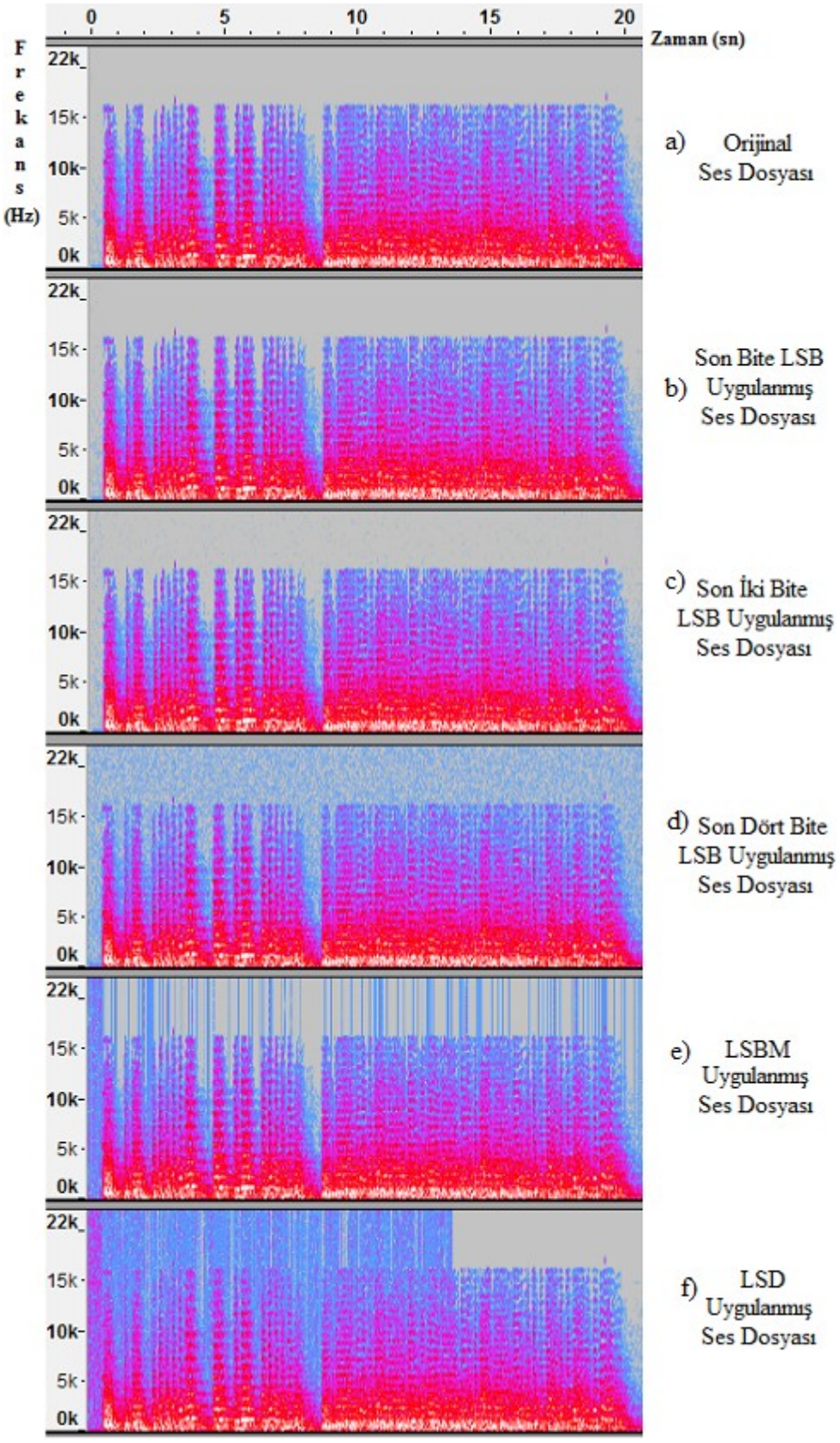
Mozart.wav dosyası üzerinde LSB ile son bit, LSB ile son iki bit ,LSB ile son dört bit, LSBM ve LSD yöntemleri kullanılarak saklama yapılmıştır. Saklama işlemleri sonucunda oluşan dosyalarda işitsel olarak algılanabilir bir gürültü meydana gelmediği gözlemlenmiştir. Bu dosyaların dalga biçimlerine incelendiğinde ise algılanabilir bir görsel değişiklik meydana gelmediği görülmüştür (Şekil 6.1).

Orijinal dosya ve saklama işlemi sonrasında oluşan dosyaların STFT (short-time Fourier transform) aracılığı ile üretilen spektogram grafikleri üretilmiştir. Bu grafikler karşılaştırıldığında (Şekil 6.2) son bit üzerinde uygulanan LSB işleminin bir fark meydana getirmediği görülmektedir (Şekil 6.2 - b) . Ancak ikinci bit ve üzerindeki bitler kullanılarak yapılan saklamanın (Şekil 6.2 – c, d) spektogramlara noktalar eklediği ve fark edilirliğe yol açtığı görülmektedir. LSB uygulanan bit sayısı arttıkça nokta miktarı artmaktadır. LSBM ve LSD yöntemleri ise grafiğe açık olarak algılanabilir düz uzun çizgiler eklemektedir (Şekil 6.2 – e,f). LSB yöntemi bit değiştirme işlemi yaparken LSBM ve LSD yöntemi sayıyı arttırmak ve azaltmak suretiyle saklama gerçekleştirmektedir. Bu yöntemler sayısal olarak düşük miktarda değişikliğe sebep oldukları halde yapılan değişiklik bayt içerisindeki bitlerin çoğunun değişmesi ihtimalini ortaya çıkarmaktadır. Bu değişimin spektogram grafiğinde çizgiler olarak ortaya çıktığı değerlendirilmektedir.

İşitsel olarak, dalga biçimi grafiği ve spektogram grafiği aracılığı ile algılanamadığı gerekçeleriyle kullanılan yöntemler içerisinde saldırılara en dayanıklı yöntemin son bit üzerinde uygulanan LSB yönteminin verdiği tespit edilmiştir.

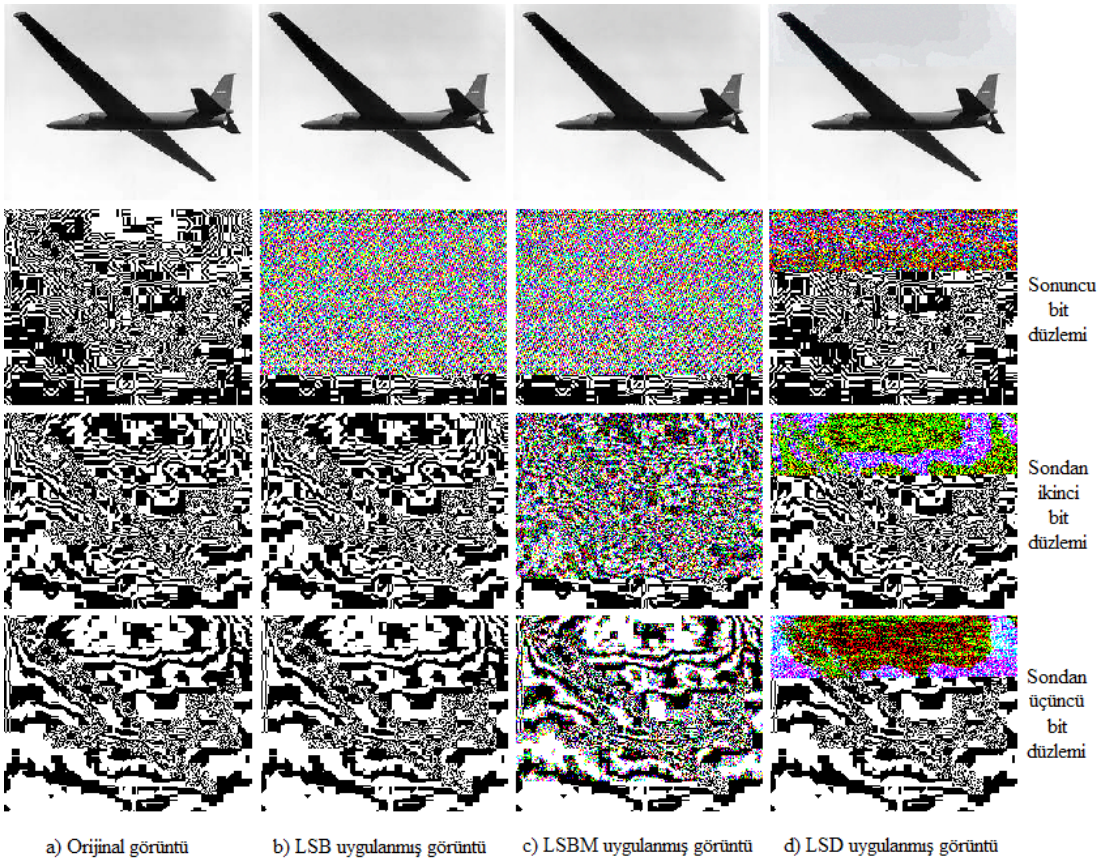


Şekil 6.1 Orijinal ve saklama yapılmış ses dosyalarının dalga biçimleri



Şekil 6.2. Orijinal ve saklama yapılmış ses dosyalarının spektogramları

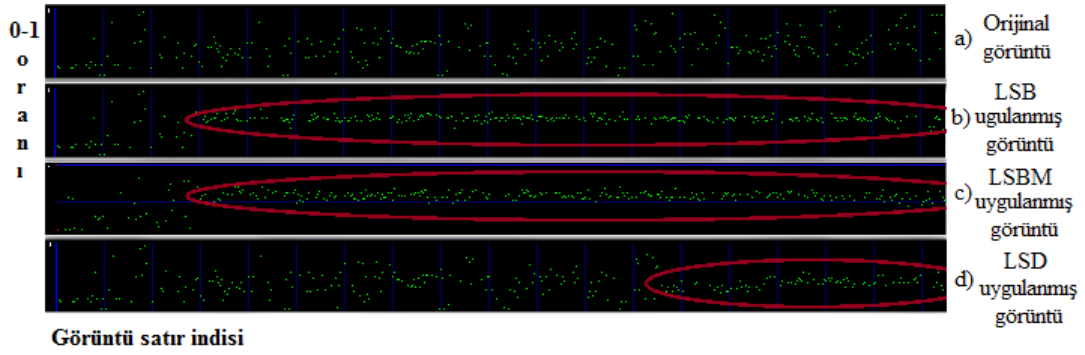
BMP görüntü üzerinde standart LSB, LSBM ve LSD kullanılarak yapılmış veri gizlemeye ait girdi ve çıktı görüntüler aşağıdadır (Şekil 6.3). Görüldüğü üzere ilk üç görüntü arasında insan gözünün algılayabileceği bir fark yoktur. Ancak dördüncü görüntüde insan gözüyle algılanabilir bir fark vardır. Bunun sebebi LSD yönteminin, 1 bayt'ın son üç bitini yoğun olarak değiştiriyor olması ve uçak görüntüsünün gri tonajlı renklerin değişkenlik göstermediği bir görüntü olmasıdır. Görüntüdeki ikinci satır son bite, üçüncü satır sondan ikinci bite, dördüncü satır ise sondan üçüncü bite görsel saldırı yapılması sonucu çıkan görüntüleri içermektedir.



Şekil 6.3. Uçak görüntüsüne görsel saldırılar

İkinci, üçüncü ve dördüncü görüntü üzerinde uygulanan steganaliz sonucunda veri gizlendiği tespit edilebilmektedir. İkinci satırda görülmektedir ki orijinal görüntünün son bit düzlemi ile LSB, LSBM ve LSD ile gizleme yapılmış görüntülerin son bit düzlemi arasında büyük fark vardır. Bu fark bu bitlerde değişiklik yapıldığını yani

veri saklandığını göstermektedir. Ayrıca bu üç yöntemle saklanan bilgiler aynı bilgilerdir. LSB ve LSBM yöntemi bir bayta bir bit saklarken LSD yöntemi üç bayta bir bayt saklama yapıldığı için değişen taralı alan üçüncü görüntüde diğerlerinin yaklaşık üç katıdır. LSB yöntemi baytın sadece ilgili bitini değiştirmektedir. LSBM yöntemi son bit düzleminde LSB yöntemi ile birebir aynı etkiyi yapmaktadır. LSBM yöntemi LSB'den farklı olarak sayı değerini bir arttırmakta veya bir azaltmaktadır. Bu işlem sayı değerinde birlik bir etkide bulunsa da üst bitlerin tümünü değiştirme ihtimaline sahiptir. Şekil 6.3'ün üçüncü sütununda LSBM yönteminin sondan ikinci ve üçüncü bitleri değiştirdiği görülmektedir. Aynı şekilde LSD yöntemi de sayının değerini dokuza kadar değiştirdiği halde tüm bitleri değiştirme ihtimaline sahiptir.



Şekil 6.4. Uçak görüntüsü, LSB, LSBM ve LSD'sine istatistiksel saldırı

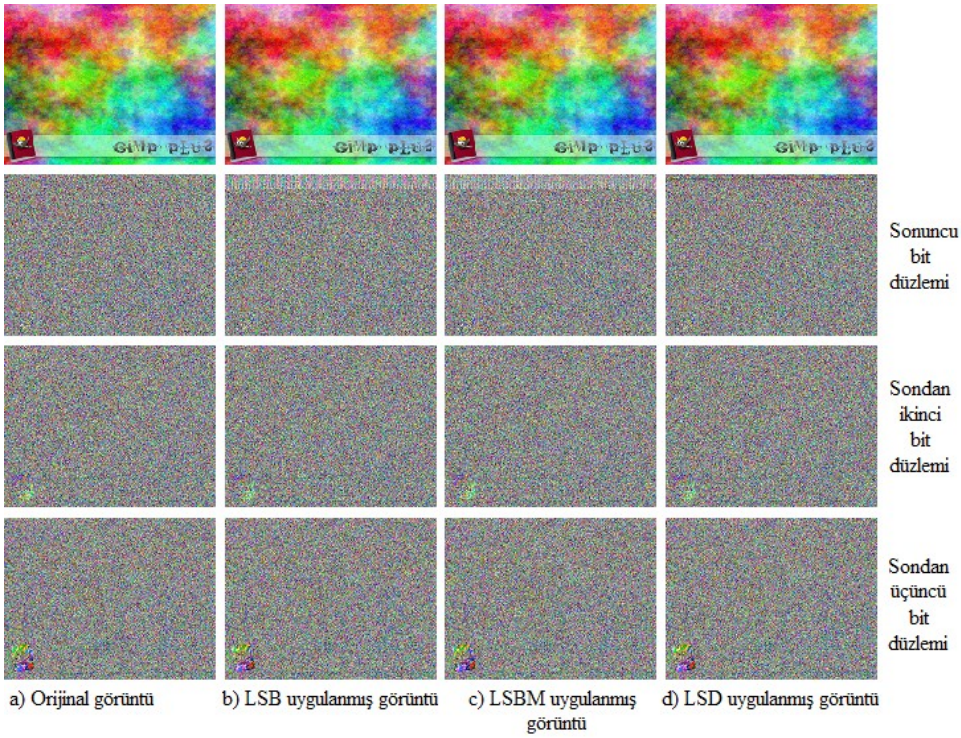
Şekil 6.4'de görüldüğü üzere uçak görüntülerine istatistiksel saldırı yapıldığında üç yöntemde de steganografi uygulandığı açığa çıkmaktadır. Son kısımdaki gri işaretli alanda 0 ve 1 dengesinin eşit olduğu gözlenmektedir. Bu saldırı sonucunda örtü görüntüde gizli bilgi olduğu söylenebilir.

Yapılan görsel ve istatistiksel saldırılara karşı PRNG kullanmayan LSBM yönteminin LSB yöntemine göre bir fark göstermediği görülmüştür. LSBM yönteminin değiştirdiği bitlerde son bitin yanı sıra sondan ikinci bitin değişme ihtimali %50'dir. Böyle bir değişim ise sondan ikinci bitten yukarı bitlerin değişimine sebep olabilir. Bu üst bitlerin değişimi her ne kadar görüntünün bozulmasına sebep olmasa da ses dosyalarında hissedilir derecede bozulmaya sebep olmaktadır.

Çizelge 6.1. Uçak PSNR ve MSE değerleri

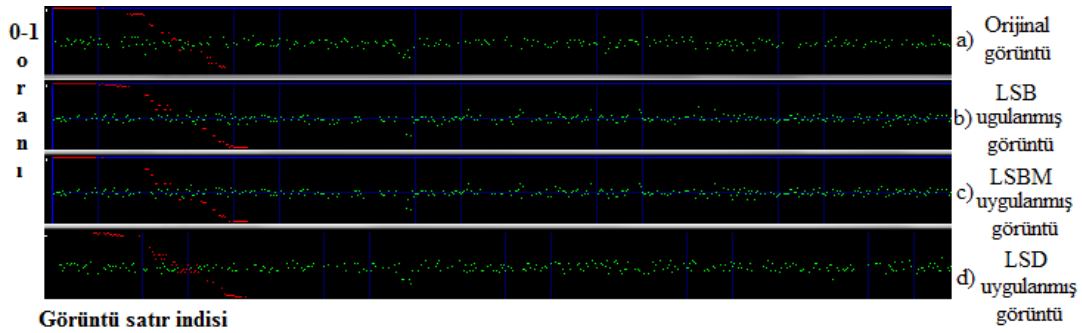
Yöntem	Orijinal Görüntü-LSB	Orijinal Görüntü-LSBM	Orijinal Görüntü-LSD	LSB-LSBM
PSNR	51.79 dB	51.79 dB	40.68 dB	48.81 dB
MSE	0.43	0.43	5.59	0.86

Çizelge 6.1'de PSNR ve MSE (Mean Squared Error) değerleri karşılaştırılmıştır. Birinci sütunda LSB ve ikinci sütunda LSBM yönteminin görüntü üzerinde eşit miktarda tahribata yol açtığı görülmektedir. Bunun sebebi bir pikseldeki renk değeri üzerinde, LSB yönteminin son biti değiştirerek yaptığı değişiklik miktarı ile LSBM yönteminin sayıyı rasgele bir arttırarak veya bit azaltarak yaptığı değişiklik miktarının aynı olmasıdır. Üçüncü sütunda LSD yönteminin eklediği gürültünün diğer iki yöntemden daha fazla olduğu görülmektedir. Dördüncü sütunda ise LSB ve LSBM yöntemleri kullanılan görüntüler arasındaki fark görülmektedir. Bu iki yöntem son bitler üzerinde aynı sonucu elde etseler de sayı değerleri birbirinden farklı olmaktadır. Bu da farkın sebebidir.



Şekil 6.5. Uçak görüntüsüne göre daha karışık olan görüntü

Renklerle dolu Şekil 6.5'de görsel saldırı kullanılınca LSB yöntemi tespit edilebilmekte iken LSD yönteminin tespit edilebilirliği çok düşüktür. Uygulanan istatistiksel saldırı da renk yoğunluğu fazla olan karmaşık görüntülerde etkisiz kalmaktadır (Şekil 6.6).



Şekil 6.6. Renkler görüntüsü ve LSB, LSD' sine istatistiksel saldırı

Çizelge 6.2'de de Çizelge 6.1'de olduğu gibi LSD yönteminin LSB ve LSBM yönteminden daha fazla gürültü eklediği görülmektedir.

Çizelge 6.2. Renkler PSNR ve MSE değerleri

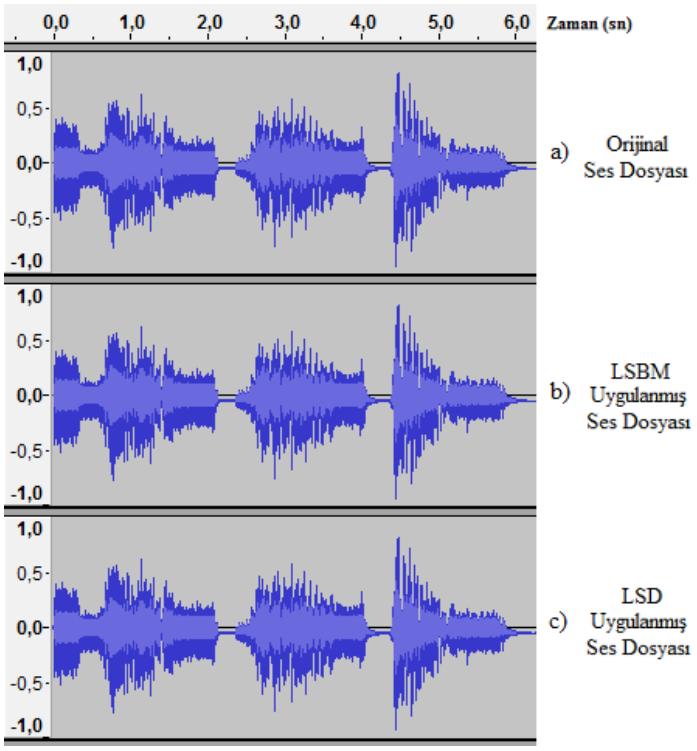
Yöntem	Orijinal Görüntü-LSB	Orijinal Görüntü-LSBM	Orijinal Görüntü-LSD	LSB-LSBM
PSNR	61.64 dB	61.64 dB	50.18 dB	58.65 dB
MSE	0.04	0.04	0.62	0.08

Uygulanan yöntemler gürültü uygulama ve sıkıştırma kullanılarak dayanıklılık açısından incelenmiştir. Tuz ve biber (Salt & Pepper), Gauss ve benekleme (Speckle) yöntemleri ile gürültü uygulanmıştır. Ayrıca görüntü Jpeg olarak sıkıştırılmış ve tekrar açılmıştır. Üç yönteminde mevcut bu işlemler sonucunda dayanıklılık sağlayamadığı görülmüştür. Dayanıklılığı arttırabilmek için PRNG kullanılarak seyrek veri saklama yapılması ve hata kontrol kodlaması uygulanması uygun olacaktır. Bu işlemlerin kapasite üzerinde olumsuz etkisi olacağı muhakkaktır.

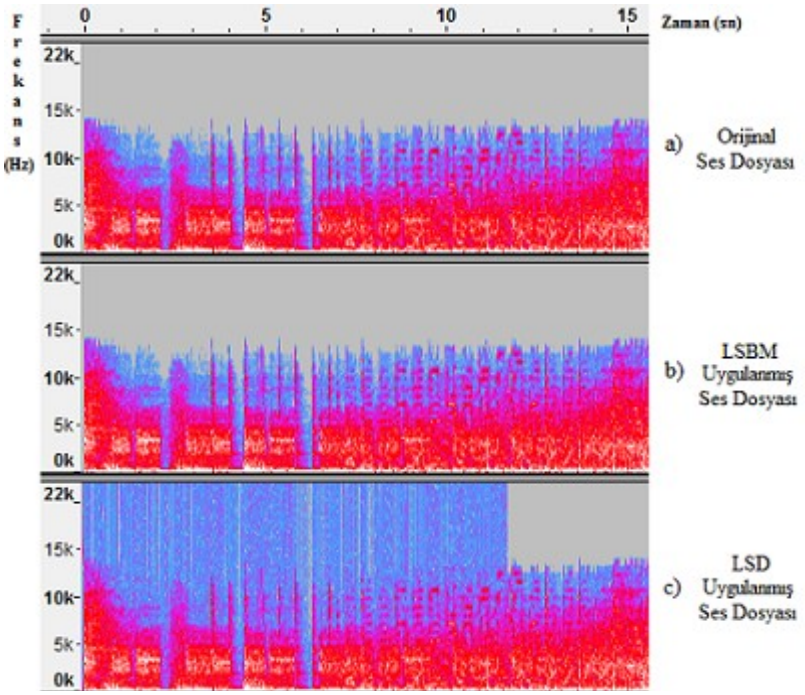
Sonuç olarak LSD yöntemi LSB'den kapasite ve güvenlik açısından daha üstündür

denilebilir. Ayrıca steganografi uygulanacak görüntünün karışık ve bol renk içeren bir görüntü seçilmesi olası saldırılara karşı güvenliği arttıracaktır.

StegoFolderLock uygulaması avi 1.0 video dosyalarının hem görüntü hem ses alanını birlikte kullanarak saklama yapabilmektedir. Örneğin 29 fps ve 352x240 çerçeve boyutuna sahip 33 saniyelik bir video dosyasını ele alacak olursak bu videonun görüntü kısmı 352x240x3x29x33 yaklaşık 230 Mbayt büyüklüğünde olacaktır. Aynı videonun 16 bit örnek boyutuyla saniyede 44100 örnekleme yapan stereo sese sahip olduğunu varsayacak olursak 44100x16/8x2x33 yaklaşık 5 Mbayt büyüklüğünde ses verisi videoda bulunacaktır. Saklama işleminde bu videonun tüm saklama kapasitesi kullanılırsa gizli bilginin yaklaşık 1/47'lik (5/230+5) bölümü videonun ses alanına, 46/47'lik (230/230+5) bölümü görüntü alanına saklanmış olacaktır. Uygulama gizli bilgiyi videonun ses ve görüntü alanlarına paket paket sakladığı için bu yönteme fermuar yöntemi de denilebilir. Bu şekilde güvenlik arttırılmaktadır. Aşağıda Stegofolderlock uygulaması ile fermuar yöntemi kullanılarak video steganografi uygulanmış dosyalar üzerinde yapılan deneyler görülmektedir.



Şekil 6.7. Orijinal ve saklama yapılmış videolardaki seslerin dalga biçimleri



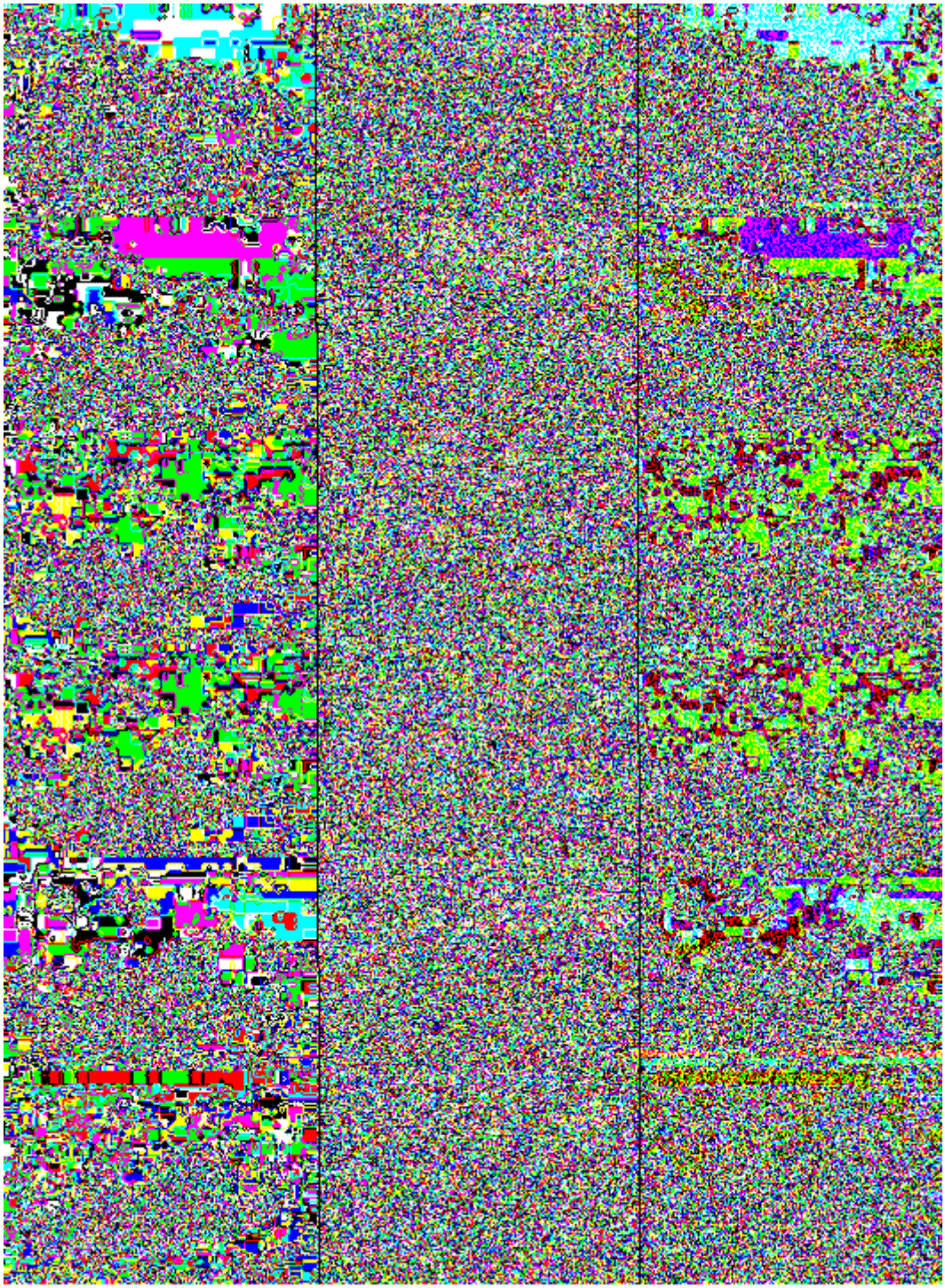
Şekil 6.8. Orijinal ve saklama yapılmış videolardaki seslerin spektogramları

Yaklaşık 250 Mbayt boyutundaki videoya LSB ve LSD yöntemleriyle saklama sonucunda oluşan videonun ses bölümünün dalga biçimi grafikleri ve spektogram grafikleri Şekil 6.7 ve Şekil 6.8'deki gibi olmaktadır. Aynı video dosyalarının görüntü alanlarından alınan çerçevelerin bir kısmı Şekil 6.9'da görülmektedir. Bu çerçevelere görsel saldırı yapıldığında alınan sonuç Şekil 6.10'daki gibi olmaktadır. Bu şekiller de göstermektedir ki yalın ses dosyası üzerinde saklamanın etkisi ile videodaki ses dosyası üzerindeki saklamanın etkisi ve video dosyalarındaki çerçeveler üzerinde uygulanan saklama ile tek bir görüntü dosyası üzerinde uygulanan saklamanın etkisi aynıdır.

Video dosyalarında steganografik açıdan fark meydan getirebilecek özellikler değerlendirilmelidir. Örneğin hareketsiz görüntüleri videodan ayıran ana özellik ardışık çerçeveler arasındaki değişimdir. Bu değişim saklama amacıyla kullanılabilir. Ardışık çerçeveler arasında değişikliğin çok olduğu bölgeler saklama açısından uygun yerlerdir. Bu şekilde insan gözünün saklamayı algılama ihtimali düşürülebilir.



Şekil 6.9. Videolardan alınmış kesit (Yukarıdan aşağı ardışık çerçeveler)



a) Orijinal video

b) LSB ile saklama yapılmış video

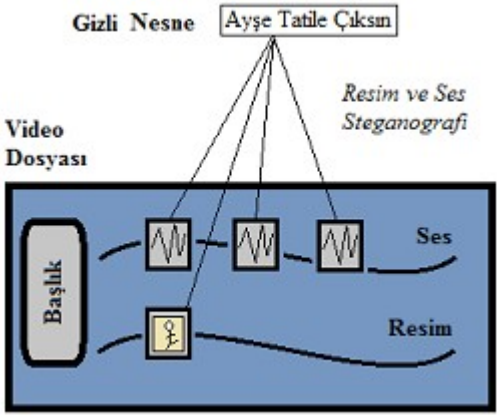
c) LSD ile saklama yapılmış video

Şekil 6.10. Videolardan alınmış kesit üzerinde steganaliz

7. SONUÇLAR VE ÖNERİLER

Bu çalışmada metin, ses, görüntü ve video steganografisi üzerine literatürde yapılan çalışmalar incelenmiştir. Veri gizleme işlemleri için kullanılacak esnek bir kütüphane tasarlanmıştır. Bu kütüphaneyi kullanan bir uygulama geliştirilmiştir. Kütüphanenin ortam ve yöntem bağımsız olmasının yanı sıra java programlama dili kullanılarak platform bağımsız olması da sağlanmıştır.

Ses ve görüntü dosyaları saklama kapasitesi açısından yetersizdir. Video dosyalarına saklama yapılarak bu kısıt aşılabilir. Kütüphanede ses, görüntü ve video dosyaları desteklenmektedir. Karmaşıklığı ve güvenliği arttırmak için video dosyasının görüntü ve ses alanları saklama işleminde birlikte kullanılmıştır (Şekil 7.1). Ayrıca kütüphanede web kamerasından çekilen videonun görüntü alanına gerçek zamanlı saklama yapılması da sağlanmıştır. Böylece asıl video dosyasıyla arasındaki farkın tespit edilmesi önlenir ve saklanacak veri miktarının ihtiyaç duyduğu ideal video çekilir.



Şekil 7.1. Güvenliği arttırmak için uygulanan saklama yöntemi

Sade seslerden oluşan ses dosyası üzerinde yapılan deney sonucunda LSB, LSBM ve LSD ile son 4 bite kadar yapılan veri gizleme işlemlerinin işitsel olarak algılanamadığı görülmüştür. Orijinal ses ve saklama yapılmış seslerin dalga boyu grafiklerine bakıldığında gözle algılanabilir bir fark meydana gelmemektedir.

Spektogram analizi sonucunda LSB yönteminin son iki bite kadar fark edilemediği ancak diğer yöntemlerin algılanabildiği görülmüştür.

Ham görüntü dosyalarında uygulanan deneyler sonucunda düz görüntülerde LSB ve LSBM yöntemlerinin gözle algılanabilir bir fark meydana getirmediği ancak LSD yönteminin algılanabilir fark meydana getirdiği görülmüştür. Düz görüntü üzerinde uygulanan üç yöntemin de son bitlere uygulanan görsel saldırı ve ki kare istatistiksel saldırısıyla tespit edilebildiği görülmüştür. Karmaşık görüntü üzerinde saklama yapılması durumunda üç yöntemin de gözle algılanamadığı ve ki kare istatistiksel saldırısının da sonuç vermediği görülmüştür. Görsel saldırı ile üç yöntemin de tespit edilebildiği ancak LSD yönteminin diğer ikisine kıyasla biraz daha dayanıklı olduğu söylenebilir.

Geliştirilen kütüphane Erkin ve Örencik'in tasarladığı kütüphaneye kıyasla saklama ortamları açısından çeşitlilik sunarken saklama yöntemleri açısından geride kalmaktadır. Erkin'in kütüphanesi sadece görüntü dosyalarında saklama yapmakta iken bu çalışmada geliştirilen kütüphane ses, görüntü ve video dosyalarını desteklemektedir. Video dosyalarının desteklenmesi ve kameradan gerçek zamanlı olarak saklama yapılabilinmesi saklama kapasitesini oldukça arttırmaktadır.

Frekans alanda çalışan yöntemlerin, adaptif yaklaşımların ve sıkıştırma yapan ortamların kütüphaneye eklenmesi güvenlik, dayanıklılık ve çeşitlilik açısından faydalı olacaktır. Gerçek zamanlı videoya saklama işleminde ses alanına da saklama desteğinin eklenmesi faydalı olacaktır. Kütüphaneye metin steganografisi paketi eklenerek güvenlik arttırılabilir ancak bu paket kapasitenin azalmasına yol açacaktır.

Yapılan çalışmada önceki çalışmalardan farklı olarak video dosyası içerisindeki görüntü alanının yanı sıra ses alanı da saklama amacıyla kullanılmış, karmaşıklık ve güvenlik arttırılmıştır. Literatürde video'ya veri saklama yapabilen uygulamaların sayısı oldukça kısıtlıdır. Bu çalışmanın ülkemizdeki videonun görüntü ve ses alanını birlikte kullanarak steganografi uygulayabilen ilk çalışma olduğu değerlendirilmektedir.

KAYNAKLAR

1. Johnson N. F., Katzenbeisser S., "A Survey of Steganographic Techniques", ***Artech House Books***, Chapter 3 (2000)

2. Cheddad A., Condell J., Curran K., PaulMcKevitt, "Digital image steganography: Survey and analysis of current methods", ***Signal Processing,*** 90, 727– 752 (2010)

3. Lashkari A. H., Abdulmanaf A., Masrom M., Daud S. M., "A Survey on Image Steganography Algorithms and Evaluation", ***ICDIPC Springer***, Czech Republic, 406-418 (2011)

4. Wang R.,Chen Y., "High-Payload Image Steganography Using Two-Way Block Matching", ***IEEE Signal Processing Letters,***13(3), 161-164 (2006)

5. Cvejic N., Seppanen T., "A Wavelet Domaın Lsb Insertion Algorithm For High Capacity Audio Steganography", ***Digital Signal Processing Workshop***, Finland, 53-55 (2002)

6. Gopalan K., Qidong Shi, "Audio Steganography using Bit Modification – A Tradeoff on Perceptibility and Data Robustness for Large Payload Audio Embedding", ***ICCCN IEEE***, USA, 1-6 (2010)

7. Wu M., Zhu Z., Jin S., "A New Steganalytic Algorithm for Detecting Jsteg", ***X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS,*** 3619, 1073 – 1082 (2005)

8. Luo X., Wang D., Wang P., Liu F., "A review on blind detection for image steganography", ***Signal Processing,*** 88, 2138– 2157 (2008)

9. Niels Provos, Peter Honeyman, "Hide and Seek: An Introduction to Steganography", ***IEEE Security & Privacy,*** 1, 32-44 (2003)

10. Gürol Canbek, Şeref Sağıroğlu, "Bilgi ve Bilgisayar Güvenliği Casus Yazılımlar ve Korunma Yöntemleri", 25-26 (2006)

11. İnternet: Turksteg uygulaması
www.teknohaber.net/resimler/bside.exe

12. İnternet: PCM görüntüsü
http://en.wikipedia.org/wiki/File:Pcm.svg

13. Kessler G. C.,"An Overview of Steganography for the Computer Forensics

Examiner", ***Forensic Science Communications***, (2004)

14. İnternet: MP3 sıkıştırma
http://www.omninerd.com/articles/How_Audio_Compression_Works

15. Noto M., "MP3Stego: Hiding Text in MP3 Files", ***SANS Institute InfoSec Reading Room***, (2001)

16. Çetin Ö., Özcerit A. T., "A new steganography algorithm based on color histograms for data embedding into raw video streams", ***Computers & Security,*** 28, 670-682 (2009)

17. İnternet: Graphics Internet Change Format,
http://www.wfu.edu/~matthews/misc/graphics/formats/formats.html

18. Keith Jack, "Video Demystified: A Handbook for the Digital Engineer 5th Edition", ***Newnes*** (2007)

19. İnternet: JPEG Steganography,
http://www.guillermito2.net/stegano/jsteg/index.html

20. Tewari B., Dubey S., M. Nizamuddin, "Compression Analysis Between DWT And DCT", **Proceedings of 2nd international Conference on IT & Business Intelligence (ITBI-10)**, India, (2010**)**

21. İnternet: JPEG 2000 Niceleme İşlemi Görüntüsü
http://www.whydomath.org/node/wavlets/jpeg2000quantization.html

22. Stanescu D., Stratulat M., Ciubotaru B., Chiciudean D., Cioarga R., Micea M, "Embedding Data in Video Stream Using Steganography", ***IEEE SACI***, Timisoara, 241-244 (2007)

23. Atıcı M. A., "Steganografik Yaklaşımların İncelenmesi, Tasarımı Ve Geliştirilmesi", ***Gazi Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği A.B.D.***, Yüksek Lisans Tezi, Ankara, 19-24 (2007)

24. Shirali-Shahreza M. H., Shirali-Shahreza M., "A New Approach to Persian/Arabic Text Steganography", ***ICIS-COMSAR'06***, USA, 310-315 (2006)

25. Samphaiboon N. and Dailey M. N, "Steganography in Thai Text", ***Proceedings of ECTI-CON 2008***, 133-136 (2008)

26. Shirali-Shahreza M. H., Shirali-Shahreza M., "A New Synonym Text

Steganography", ***International Conference on Intelligent Information Hiding and Multimedia Signal Processing***, Iran, 1524 - 1526 (2008)

27. Sun X., Meng P., Ye Y., Hang L., "Steganography in Chinese Text", ***ICCASM IEEE***, China, 651-654 (2010)

28. Meral H. M., Sankur B., Özsoy A. S., "Türkçe metin belgeleri için damgalama", ***Signal Processing and Communications Applications***, İstanbul, 1-4 (2006)

29. Asad M., Gilani J., Khalid A., "An Enhanced Least Significant Bit Modification Technique for Audio Steganography", ***ICCNIT IEEE***, Pakistan, 143-147 (2011)

30. Yan D., Wang R., "Huffman table swapping-based steganography for MP3 audio", ***Multimedia Tools and Applications***, 52(2-3), 291-305 (2009)

31. Shirali-Shahreza M.H., Shirali-Shahreza S., "Real-time and MPEG-1 layer III compression resistant steganography in speech", ***Information Security IET***, 4(1), 1-7 (2010)

32. Westfeld A., Pfitzmann A., "Attacks on Steganographic Systems", ***Lecture Notes on Computer Science,*** 1768 (2000)

33. Guillermo A. Francia, III, Tyler S. Gomez, "Steganography Obliterator: An Attack on the Least Significant Bits", ***InfoSecCD '06*** (2006)

34. Bender W., Gruhl, D., Morimoto, N., Lu, A., "Techniques for data hiding", ***IBM Systems Journal***, 313-336 (1996)

35. Yeo I., Kim H. J., "Modified Patchwork Algorithm: A Novel Audio Watermarking Scheme", ***Information Technology: Coding and Computing***, 237-242 (2003)

36. Yeo I., Kim H. J., "Generalized patchwork algorithm for image watermarking", ***Multimedia Systems*** ***Springer-Verlag***, 9(3), 261-265 (2003)

37. Sharp T., "An Implementation of Key-Based Digital Signal Steganography", ***IH 2001 LNCS,*** 2137, 13-26 (2001)

38. Mielikainen J., "LSB Matching Revisited", ***IEEE Signal Processing Letters***, 13(5), 285-287 (2006)

39. Luo W., Huang F., Huang J., "Edge Adaptive Image Steganography Based on LSB Matching Revisited", ***Information Forensics and Security***, 201-214

(2010)

40. Mathkour H., Assassa G. M. R., Al Muharib A., Kiady I., "A Novel Approach for Hiding Messages in Images", ***International Conference on Signal Acquisition and Processing***, Saudi Arabia, 89-93 (2009)

41. Akar F, Varol H. S., "A new RGB weighted encoding technique for efficient information hiding in images", ***Journal of Naval Science and Engineering***, 2, 21–36 (2004)

42. Chen S., Wang R., "High-payload image hiding scheme using k-way block matching", IIH-MSP, Taiwan, 70-73 (2010)

43. T. Morkel, J.H.P. Eloff, M.S. Olivier, "An Overview of Image Steganography", ***In Proceedings of the ISSA 2005 New Knowledge Today Conference,*** South Africa, 1-11 (2005)

44. Marvel L. M., Boncelet C. G., Retter C. T., "Spread Spectrum Image Steganography", ***Image Processing***, 1075 - 1083 (1999)

45. K. Satish, T. Jayakar, Tobin C., K. Madhavi and K. Murali, "Chaos Based Spread Spectrum Image Steganography", ***Consumer Electronics,*** 587-590 (2004)

46. Youail R. S., Samawi V. W., Kadhim A. A-R., "Combining a Spread Spectrum Technique with Error-Correction Code to Design an Immune Stegosystem", ***Anti-counterfeiting, Security and Identification ,*** 245-248 (2008)

47. Agrawal N., Gupta A., "DCT Domain Message Embedding In Spread-Spectrum Steganography System", Data Compression Conference **,** 433 (2009)

48. Shirali-Shahreza M., "Steganography in MMS", **IEEE,** Tehran, 1-4 (2007)

49. Seppänen T., Mäkelä K., Keskinarkaus A., "Hiding Information in Color Images Using Small Color Palettes", ***ISW LNCS***, 69-81 (2000)

50. G. Brisbane, R. Safavi-Naini, P. Ogunbona, "High-capacity steganography using a shared colour palette", ***IEEE Proc.-Vis. Image Signal Process.***, Australia, 152(6), 787-792 (2005)

51. İnternet: JSteg yazılımı http://www.jsteg.org

52. P. Wayner, "Disappearing Cryptography second edition", ***Morgan Kaufmann Publishers*** (2002)

53. Westfeld A., "F5—A Steganographic Algorithm High Capacity Despite Better Steganalysis", ***S. Moskowitz (Ed.): IH 2001, LNCS,*** 2137, 289–302 (2001)

54. Fridrich J., Goljan M., Hogea D., "Steganalysis of JPEG Images: Breaking the F5 Algorithm", ***F.A.P. Petitcolas (Ed.): IH 2002, LNCS,*** 2578, 310-323 (2003)

55. Noda H., Niimi M, Kawaguchi E., "Application Of QIM With Dead Zone For Histogram Preserving Jpeg Steganography", ***ICIP IEEE,*** 1082-1087 (2005)

56. Chang C., Chen T., Chung L, "A steganographic method based upon JPEG and quantization table modification", ***Information Sciences,*** 141, 123-128 (2002)

57. Li X., Wang J., "A steganographic method based upon JPEG and particle swarm optimization algorithm", ***Information Sciences,*** 177, 3099–3109 (2007)

58. Famili Z., Faez K., Fadavi A., "A New Steganography Based on $\chi 2$ Technic", ***Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications,*** 5856, 1062-1069 (2009)

59. Liu Q, Sung A. H., Chen Z., Huang X., "A JPEG-Based Statistically Invisible Steganography", ***ICIMCS' 11***, 78-81 (2011)

60. Abdelwahab A. A., Hassaan L. A., "A Discrete Wavelet Transform Based Technique For Image Data Hiding", ***Radio Science Conference,*** Cairo, 1-9 (2008)

61. Kumar V., Kumar D., "Performance Evaluation of DWT Based Image Steganography", ***Advance Computing,*** India, 223-228 (2010)

62. Lee S., Yoo C. D., Kalker T., "Reversible Image Watermarking Based on Integer-to-Integer Wavelet Transform", ***Information Forensics and Security***, 321- 330 (2007)

63. Seyedi S. H., Aghaeinia H., Sayadian A., "A New Robust Image Adaptive Steganography Method in Wavelet Domain", ***Electrical Engineering,*** Tehran, 1-5 (2011)

64. Eltahir M. E., Kiah M. L. M., Zaidan B. B., Zaidan A. A., "High Rate Video Streaming Steganography", ***IEEE ICIME,*** Kuala Lumpur, 550-553 (2009)

65. Xu C., Ping X., Zhang T., "Steganography in Compressed Video Stream",

IEEE ICICIC, Beijing, 269-272 (2006)

66. Hussein A. Aly, "Data Hiding in Motion Vectors of Compressed Video", ***IEEE***, Cairo, 14-18 (2011)

67. Sherly A. P. and Amritha P. P., "A Compressed Video Steganography using TPVD", ***International Journal of Database Management Systems (IJDMS)***, India, 67-80 (2010)

68. İnternet: Steganografik Uygulamalar
www.jjtc.com/Steganography/tools.html

69. İnternet: OpenPuff yazılımı
http://embeddedsw.net/OpenPuff_Steganography_Home.html

70. İnternet: MSU StegoVideo yazılımı
http://compression.ru/video/stego_video/index_en.html

71. Z. Erkin, B. Örencik, "Steganografik Kütüphane", ***Ağ ve Bilgi Güvenliği Ulusal Semp. (ABG 2005) Bildiriler Kitabı***, İstanbul, 98-102 (2005)

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÜNLÜ, Onur
Uyruğu : T.C.
Doğum Tarihi ve Yeri : 27.09.1985, Ankara
Medeni Hali : Evli
E-posta : unlu00@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Yılı
Lisans	Gazi Üni. Bilgisayar Müh.	2008
Lise	Gölbaşı Anadolu Lisesi	2003

İş Deneyimi

Yıl	Yer	Görev
2010- ….	Tübitak G222	Yazılım Mühendisi
2009-2010	Yüksek Seçim Kurulu	Çözümleyici
2008-2009	LST Yazılım	Yazılım Mühendisi

Yabancı Dil

İngilizce, İspanyolca

Hobiler

Masa Tenisi, Gezi