**T.C.**
**TRAKYA ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**

# GÖRÜNTÜ STEGANOGRAFİDE KULLANILAN YENİ METODLAR VE BU METODLARIN GÜVENİLİRLİKLERİ

**Andaç ŞAHİN**

**Doktora Tezi**

**Bilgisayar Mühendisliği Anabilim Dalı**
**Danışman: Yrd. Doç. Dr. Ercan BULUŞ**

**2007**
**EDİRNE**

Doktora Tezi
Trakya Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Bölümü

# ÖZET

Bu tez bilgi gizlemenin önemli bir alt dalı olan steganografi ile ilgilidir. Tezin amacı, görüntü dosyaları içerisine bilgi gizlemek için kullanılan steganografik metotların incelenmesi ve bu yöntemlerin güvenilirliklerinin ölçülmesidir. Bu amaç doğrultusunda görüntü steganografide kullanılan yöntemler ayrıntılı bir biçimde incelenmiş ve yaygın olarak kullanılan Son Bite Ekleme Yöntemine göre sayısal resmin içerisine sıralı bir şekilde bilgi gizleyen bir uygulama geliştirilmiştir. Bu uygulama, steganografik sistem değerlendirme kriterleri olan taşıyıcıdaki değişim, kapasite ve dayanıklılık ölçütlerine göre değerlendirilmiştir. Uygulamanın dayanıklılık kriterine göre test edilebilmesi amacıyla bir steganaliz uygulaması geliştirilerek programın güvenilirliği incelenmiştir.

Tezin ilk bölümünü olan giriş bölümünde bilgi gizleme kavramı açıklanmış; kullanıldığı alanlar, kullanım amaçları ve alt alanları ayrıntılı bir şekilde anlatılmıştır.

İkinci bölümde bilgi gizlemenin önemli bir alt disiplini olan steganografi konusu incelenmiş; tanımı, tarihçesi ve metin, görüntü ve ses dosyaları üzerinde kullanım yöntemleri açıklanmıştır.

Üçüncü bölümde öncelikle, sayısal ortamda bulunan görüntü dosyalarının özellikleri açıklanmış ve görüntü dosyalarına veri gizleyebilmek için kullanılan steganografik yöntemler anlatılmıştır.

Dördüncü bölümde görüntü dosyaları içerisine bilgi gizlemek için yaygın olarak kullanılan Son Bite Ekleme metoduna göre çalışan algoritmalar ve programlar anlatılmıştır.

Beşinci bölümde bir steganografik sistemin değerlendirilmesi için gerekli kriterler anlatılmış ve bir resmin içinde bilgi olup olmadığını anlamak için kullanılan steganaliz yöntemleri incelenmiştir.

Son bölüm olan altıncı bölümde ise Sıralı LSB yöntemine göre çalışan Stego_LSB isimli bir uygulama geliştirilmiştir. Ayrıca bu programın steganografik sistemin değerlendirme kriterlerine göre incelenebilmesi için de, iki steganaliz yöntemini kullanarak işlem yapabilen bir uygulama geliştirilmiştir. Stego_LSB isimli program taşıyıcıdaki değişim, kapasite ve dayanıklılık ölçütlerine göre değerlendirilerek güvenilirliği incelenmiştir.

Anahtar Kelimeler: Bilgi Gizleme, Steganografi, Görüntü Steganografi, Son Bite Ekleme Yöntemi, Steganaliz

Yıl: 2007
Sayfa: 106

Doctorate Thesis
Trakya University Graduate School of
Natural and Applied Sciences
Department of Computer Engineering

# ABSTRACT

This thesis is related with one of the important branch of information hiding technique known as Steganography. The aim of the thesis to examine steganographic techniques used for hiding information on image files and to measure reliability of these techniques. To realize our aim, we examined the techniques used in hiding information on image files in detail. Moreover, we developed an application hiding information in arranged order on images according to commonly used LSB method. This application is evaluated according to the evaluation criteria in steganographic systems like change on cover object, capacity and robustness. To test our application according to the robustness criterion, we developed a steganalysis application to examine the reliability of our computer program.

In the first chapter of the thesis being introductory chapter we examined the concept of hiding information , the fields in hiding information that are used, the aim of usage of information hiding techniques and sub categories of information hiding techniques in detail.

In the second chapter, steganography which is sub discipline of information hiding is examined in the following subjects: introduction, history and the usage methods on text, image and audio files.

In the third chapter, the properties of digital image files are explained and Steganographic methods that are used to hide information are discussed.

In the fourth chapter, algorithms and computer programs which use LSB method as an information hiding technique are discussed.

In the fifth chapter, necessary criteria for the evaluation of steganographic systems are discussed and steganalysis methods to discover information whether it is on image or not are examined.

In the last chapter, an application (a computer program), called Stego-LSB, which use LSB method to hide information in arranged order is developed. In addition, an application for the two steganalysis method to examine this computer program according to steganographic system criteria is developed. Moreover, the reliability of the Stego-LSB program is examined according to the change on cover object, capacity and security criteria.

Key Words: Information Hiding, Steganography, Image Steganography, Least Significant Bit Insertion Method, Steganalysis

Year: 2007
Page: 106

# TEŞEKKÜR

Bu çalışmanın hazırlanması esnasında bana yol gösteren, bu alanda çalışmam için beni teşvik eden, yardımlarını ve desteğini benden esirgemeyen değerli danışman hocam Yrd. Doç. Dr. Ercan BULUŞ'a teşekkür ederim.

Tez izleme komitesinde bulunan ve değerli yorumlarıyla tez çalışmama yaptıkları katkılardan dolayı Yrd. Doç. Dr. Aydın CARUS, Yrd. Doç. Dr. Şaban AKTAŞ'a, tez jürisinde yer alan Prof. Dr. Mesut RAZBONYALI, Yrd. Doç. Dr. Rembiye KANDEMİR ve Yrd. Doç Dr. Akif SABANER'e teşekkürlerimi sunarım.

Çalışmalarım sırasında değerli katkılarıyla bana yardım eden ve ortak çalışmalar yaptığımız arkadaşlarım Arş. Gör. Dr. M. Tolga SAKALLI ve Arş. Gör. H. Nusret BULUŞ'a, çalışabilmem için gerekli ortamı sağlayan tüm mesai arkadaşlarıma çok teşekkür ederim.

Son olarak manevi desteğini benden esirgemeyen ve her zaman yanımda olan Öğr. Gör. Dr. Altan MESUT'a ve aileme teşekkür ederim.

# İÇİNDEKİLER

**ÖZET** ..... **i**

**ABSTRACT** ..... **iii**

**TEŞEKKÜR** ..... **v**

**İÇİNDEKİLER** ..... **vi**

**KISALTMALAR LİSTESİ** ..... **x**

**1. GİRİŞ** ..... **1**

1.1. Bilgi Gizleme ..... 2
1.2. Bilgi Gizlemenin Alt Alanları ..... 3
1.2.1. Gizli Kanallar (Covered Channels) ..... 3
1.2.2. Gerçek Kimliği Saklama (Anonymity) ..... 4
1.2.3. Telif Hakkı İşaretlemesi (Copyright Marking) ..... 4
1.2.4. Steganografi (Steganography) ..... 6

**2. STEGANOGRAFİ** ..... **7**

2.1. Steganografi Nedir? ..... 7
2.2. Steganografinin Tarihçesi ..... 9
2.3. Steganografinin Alt Alanları ..... 12
2.4. Steganografinin Kullanım Alanları ..... 13
2.4.1. Metin (text) Steganografi ..... 14
2.4.1.1. Açık alan yöntemleri ..... 14
2.4.1.2. Yazımsal yöntemler ..... 16
2.4.1.3. Anlamsal yöntemler ..... 17
2.4.2. Görüntü (Image) Steganografi ..... 17
2.4.3. Ses (Audio) Steganografi ..... 18
2.4.3.1. Düşük bit kodlaması ..... 19
2.4.3.2. Aşama kodlaması ..... 20

2.4.3.3. Taft yayılması....20
2.4.3.4. Yankı veri gizlemesi....20
2.4.4. Kullanılan Diğer Ortamlar....21

**3. GÖRÜNTÜ (IMAGE) STEGANOGRAFİ....23**

3.1. Sayısal Resmin Yapısı....24
3.2. Resim Dosyalarının Sıkıştırılması....26
3.3. Veri Gömme İşlemi....26
3.4. Veri Gömme Yöntemleri....28
3.5. Görüntü Dosyalarında Steganografik Yöntemler....29
3.5.1. Patchwork Algoritması....30
3.5.2. Amplitude (Bolluk) Modülasyonu Kullanılarak Bilgi Gizleme....32
3.5.3. Superposition Algoritması....33
3.5.4. SSIS (Spread Spectrum Image Steganography) Yöntemi....34
3.5.5. Frekans Domaini İçine Veri Saklanması....35
3.5.6. Son Bite Ekleme (Least Significant Bit Insertion-LSB) Yöntemi....36
3.5.6.1. Gri seviye resimler üzerinde LSB yönteminin uygulanması....36
3.5.6.2. 8-bit Renkli Resimler ve LSB yönteminin uygulanması....37
3.5.6.3. 24-bit Renkli Resimler ve LSB yönteminin uygulanması:....38
3.5.7. Steganografik Yazılımlar....39
3.5.7.1. Outguess....39
3.5.7.2. Stego Machine....39
3.5.7.3. bmpSteg....39
3.5.7.4. Stella....40
3.5.7.5. SecureEngine Professional....40
3.5.7.6. Hermetic Stego....40
3.5.7.7. GifShuffle....40
3.5.7.8. Revelation....41
3.5.7.9. Deogol....41
3.5.7.10. InfoStego....41

**4. LSB YÖNTEMİNİ KULLANAN ALGORİTMALAR VE PROGRAMLAR....42**

4.1. EzStego Algoritması ..... 43
4.2. S-Tools ..... 45
4.3. Hide and Seek ..... 46
4.4. J-Steg ..... 47
4.5. F5 Algoritması ..... 51
4.6. Ayrık Logaritma Kullanan Rasgele LSB Ekleme Yöntemi ..... 53

**5. STEGANOGRAFİK SİSTEMİ DEĞERLENDİRME KRİTERLERİ ..... 59**

5.1. Taşıyıcıdaki Değişim ..... 59
5.2. Kapasite ..... 60
5.3. Dayanıklılık ..... 60
5.3.1 $\chi^2$ Testi ..... 62
5.3.2. Histogram Analizi (PoVs'lerin Analizi) ..... 64
5.3.3. RS Steganaliz (İkili İstatistik Yöntemi) ..... 66
5.3.4. RQP Yöntemi ..... 69
5.3.5. Görsel Ataklar ..... 70
5.3.6. JPEG Steganaliz ..... 71
5.3.6.1. Orijinal ve Bilgi Gizli Resmin Elimizde Olduğu Durum ..... 72
5.3.6.2. Sadece Bilgi Gizli Resmin Elimizde Olduğu Durum ..... 74

**6. LSB YÖNTEMİ KULLANILARAK GELİŞTİRİLEN Stego_LSB PROGRAMI VE DEĞERLENDİRİLMESİ ..... 76**

6.1. 24-bit Renkli Resimler Üzerinde Sıralı LSB Uygulaması ..... 76
6.2. Geliştirilen Uygulamanın Değerlendirilmesi ..... 79
6.2.1. Taşıyıcıdaki Değişim Açısından Değerlendirilmesi ..... 79
6.2.2. Kapasite Açısından Değerlendirilmesi ..... 81
6.2.3. Dayanıklılık Açısından Değerlendirilmesi ..... 81
6.2.3.1. RS Steganaliz Uygulaması ..... 82
6.3.3.2. RQP Steganaliz Uygulaması ..... 87
6.3.3.3. Histogram Analizi ..... 90

**7. SONUÇLAR ..... 93**

**KAYNAKLAR** ..... **96**

**TEZ SIRASINDA YAPILAN ÇALIŞMALAR** ..... **104**

Ulusal Hakemli Dergi Makaleleri ..... 104
Uluslararası Kongre ve Sempozyum Bildirileri ..... 104
Ulusal Kongre ve Sempozyum Bildirileri ..... 104

**ÖZGEÇMİŞ** ..... **106**

# KISALTMALAR LİSTESİ

| | |
|---|---|
| 3DES | Triple Data Encryption Standard |
| AES | Advanced Encryption Standard |
| AIIF | Audio Interchange File Format |
| BMP | Windows Bitmap |
| CBC | Cipher Block Chaining |
| DCT | Discrete Cosine Transform |
| DES | Data Encryption Standard |
| DFT | Discrete Fourier Transform |
| DOS | Disc Operating System |
| GIF | Graphics Interchange Format |
| HAS | Human Auditory System |
| HTML | HyperText Markup Language |
| IDEA | International Data Encryption Algorithm |
| IP | Internet Protocol |
| JPEG | Joint Photographic Experts Group |
| LSB | Least Significant Bit |
| MD5 | Message-Digest Algorithm 5 |
| MPEG | Moving Picture Experts Group |
| MSE | Mean Squared Error |
| PAE | Peak Absolute Error |
| PCBC | Propagating Cipher Block Chaining |
| PNM | Portable aNy Map |
| PNG | Portable Network Graphics |
| PoVs | Pairs of Values |
| PSNR | Peak SNR (Signal-to-Noise Ratio) |
| RQP | Raw Quick Pairs |
| RMSE | Root MSE (Mean Squared Error) |
| WAV | Windows Audio-Visual |

# 1. GİRİŞ

Son yıllarda bilgisayar sistemlerinin güvenliği ve özellikle bilgi güvenliği oldukça önemli bir konu olarak karşımıza çıkmaktadır. Özellikle son 10 yılda internetin yaygınlaşmasıyla veri alışverişi ve paylaşımı da artmıştır. Metin, resim, ses vb. gibi birçok veriyi içeren dosyalar, etkin bir şekilde dünyanın birçok yerindeki insanlar tarafından paylaşılabilir hale gelmiştir. Fakat hayatı kolaylaştıran bu iletişim ağı çok ciddi güvenlik açıklarını da beraberinde getirmiştir. Birbiriyle haberleşen iki kişi arasındaki iletişim bir üçüncü kişi tarafından erişilebilir ve değiştirilebilir hale gelmiştir.

Bunu engellemek amacıyla çeşitli koruma mekanizmaları geliştirilmiş ve yeni teknolojiler ve yeni uygulamalar ortaya çıkmıştır. Bu teknolojilerden biri şifrelemedir (kriptoloji). Şifrelemede gönderilecek ve korunması istenen sayısal veri şifreleme algoritmalarıyla bir anahtar yardımıyla anlaşılmaz bir hale dönüştürülür ve bu şekilde gönderilir. Ancak şifrelerin de zaman içinde kırılabilmesi şifreleme güvenli iletişim için tek başına yeterli olmadığını göstermektedir. Bu nedenle şifreleme ve bilgi gizleme (information hiding) yöntemleri, özellikle de steganografi, birlikte kullanılarak güvenli bir iletişimin yapılması sağlanabilmektedir.

Bilgi gizleme iletişim güvenliği için oldukça önemli bir konudur. Bilgi gizlemede amaç iletişimin bir üçüncü kişinin fark edemeyeceği şekilde yapılmasıdır. Şifrelemede üçüncü kişi gizli bir bilginin gönderildiğinden haberdardır, fakat bilgi gizleme yöntemleriyle iki kişi arasındaki iletişimin gizli bir şekilde yapılması mümkün olmaktadır. Üçüncü kişi arada gizli bir iletişim olduğunu fark edememektedir.

Bilgi gizleme çok eski yıllardan beri kullanılmaktadır. Günümüzde teknolojinin gelişmesiyle birlikte birçok yeni teknik geliştirilmiştir ve hala geliştirilmeye devam edilmektedir [Katzenbeisser ve Petitcolas, 2000]. Bilgi gizlemenin çok önemli bir alt disiplini olan *Steganografi*, sayısal (dijital) ortamdaki verilerin (metin, ses ve görüntü dosyaları) korunması için son yıllarda sıklıkla kullanılmaktadır. *Steganaliz* ise gizli yapılan iletişimin ele geçirilmesi için yapılan saldırıları içermektedir.

Bir steganografik sistemin güvenilirliği çeşitli açılardan değerlendirilmektedir. Bunlar bilgi gizlemenin örtü verisini (cover object) ne kadar değiştirdiği, bilgi saklama kapasitesinin ne kadar olduğu ve dayanıklılığının ne kadar olduğudur. Dayanıklılık ölçütü steganalitik yöntemlere karşı ne kadar başarılı olduğu ile ölçülmektedir.

Teknolojinin gelişmesiyle birlikte birçok steganografik yöntem ortaya çıkmıştır, bu gelişmeyle birlikte birçok steganalitik yöntemin de geliştirilmesi gerekmiştir. Her steganografik yöntem farklı bir metod izlediği için bunları sezmede kullanılacak steganalitik yöntemler de çeşitlidir. Bir steganografik yöntem için geliştirilen steganaliz yöntemi bir diğeri için çalışmamaktadır. Her steganografik yöntemin kendine özgü bir steganaliz yöntemi bulunmaktadır.

## 1.1. Bilgi Gizleme

Bilgi gizleme iki kişi arasında yapılan iletişimin bir üçüncü kişi tarafından fark edilmeyecek şekilde gerçekleştirilmesidir. Bilgi gizlenmenin amacı, iletişimimizdeki veriyi veya iletişiminin amacını saklamaktır.

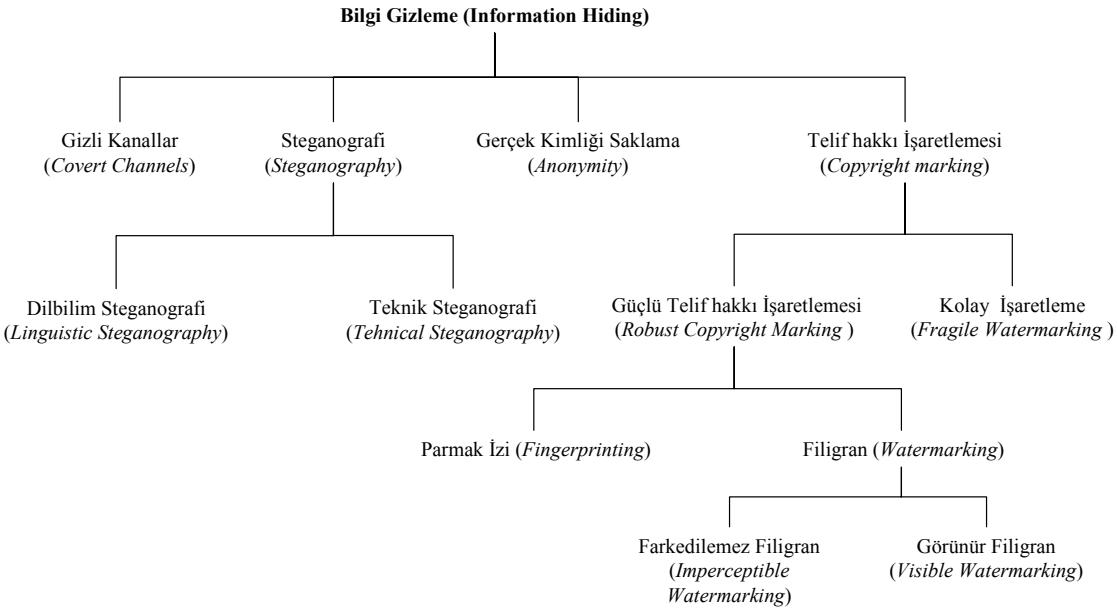
Bilgi gizleme, gönderilecek bilginin önemli olduğu birçok alanda sıklıkla kullanılmaktadır.

Askeri ve istihbarat birimleri, aralarındaki iletişimin güvenli olmasını, gönderdikleri gizli verilere başka kişilerinin erişmemesini isterler. İçerik şifrelense bile, modern savaş sahalarında bir sinyalin yakalanmasıyla saldırgan bu sinyale saldırabilir ve bilgiyi ele geçirebilir. Bu nedenle, askeri iletişimlerde sinyalin düşman tarafından bulunmasını zorlaştıran çeşitli bilgi gizleme teknikleri kullanılmaktadır.

Suçlular da kendi aralarındaki iletişimin güvenlik güçleri tarafından anlaşılmaması için çeşitli yöntemler kullanmaktadırlar. Örneğin cep telefonları frekanslarını kendi aralarındaki iletişim için düzenleyebilmekte ve diğer kişiler bunları duyamamasını sağlamaktadırlar. Buna karşılık olarak güvenlik güçleri de bu iletişimi ele geçirebilmek için karşı yöntemler geliştirmiştir [Petitcolas vd., 1999].

## 1.2. Bilgi Gizlemenin Alt Alanları

İletişimin gizli bir şekilde gerçekleştirilebilmesi için çeşitli yöntemler geliştirilmiştir. Bilgi gizlemenin sınıflandırılması Şekil 1.1.'de gösterilmektedir [Pfitzmann, 1996]. Bu sınıflandırma, bilgi gizleme konusunda yapılan ilk çalıştay'da üzerinde çalışılarak kabul edilmiştir [Anderson, 1996].




**Şekil 1.1.** Bilgi gizleme yöntemlerinin sınıflandırılması

### 1.2.1. Gizli Kanallar (Covered Channels)

Bilgi gizlemenin ilk alt disiplini olan gizli kanallar Lampson [Lampson, 1973] tarafından tanımlanmıştır. Gizli kanallar iki kişi arasında gizli bilgilerin el değiştirmesi için iletişimi sağlayan kanaldır.

Gizli kanal kurulması iki kişinin karşılıklı anlaşmasını gerektirmektedir. Gizli kanalların amaçları, iletişimimizdeki veriyi saklamaya çalışmak ve iletişiminin amacını saklamaktır. Böylece; gerçek veri transferi, dikkatsiz gözlere zararsız ve kanuna uygunmuş gibi gözükecek ve veriyi karıştırmak için ayrı bir şifreleme yapılmasına gerek kalmayacaktır.

### 1.2.2. Gerçek Kimliği Saklama (Anonymity)

Diğer bir alt alan olan gerçek kimliği saklama, veri gönderimi sırasında gerçek kimliği saklayarak, bilginin bilinmeyen ya da anlaşılamayan biri üzerinden gidiyor olduğunu izlenimi verilerek gönderilmesidir. Bu şekilde bilgi zarar görmeden gönderilebilmektedir. Fakat ağlar üzerinde bilinmeyen kullanıcı olayı ağ yöneticilerinin daha fazla dikkatini çekmekte ve bilgi güvenliği tehlikeye girmektedir. Bu yüzden sadece çok gerektiği durumlarda kullanılması uygundur [Chaum, 1981] [Goldschlag vd.,1996].

### 1.2.3. Telif Hakkı İşaretlemesi (Copyright Marking)

Telif hakkı işaretlemesinde ise orijinal dosyanın korunması amacıyla dosyanın içine bazı bilgiler gizlenmektedir. Bunlar; dosyaların üretildiği tarih, telif hakkı sahibi, üreticiye nasıl ulaşılabileceği gibi bilgileri içermektedir. Bu yöntemler steganografi ile beraber kullanılmaktadır. Telif hakkı işaretlemesi, sayısal görüntülerde sayısal filigran olarak kullanılmaktadır [Swanson vd., 1998]. Filigran, bir çeşit gizli damga baskısıdır. Örneğin kâğıt banknotlar üzerindekiler gibi. Bunlar ancak ışığa tutularak bakıldıklarında görülebilmektedirler.

Modern steganografi uygulamalarında kullanılan filigranlar ise görüntü ve ses dosyalarında kopyalamayı önlemek amacıyla damgalar bırakmaktadırlar [Hartung ve Kutter, 1999]. Bu damgalar özel programlar tarafından okunabilmekte ve dosyaların üretildiği tarih, telif hakkı sahibi, üreticiye nasıl ulaşılabileceği gibi bilgileri içermektedir.

Filigran ile korunmuş görüntüler parlaklık ve zıtlık (kontrast) ayarlarının değiştirilmesi, özel filtrelerin kullanılması, kâğıda baskı ve tarama gibi birçok yönteme karşı koyabilmektedir. Fakat gelişen teknolojiyle birlikte ortaya çıkartılan bazı yeni programlar kullanılarak bu filigran aşılabilmektedir.

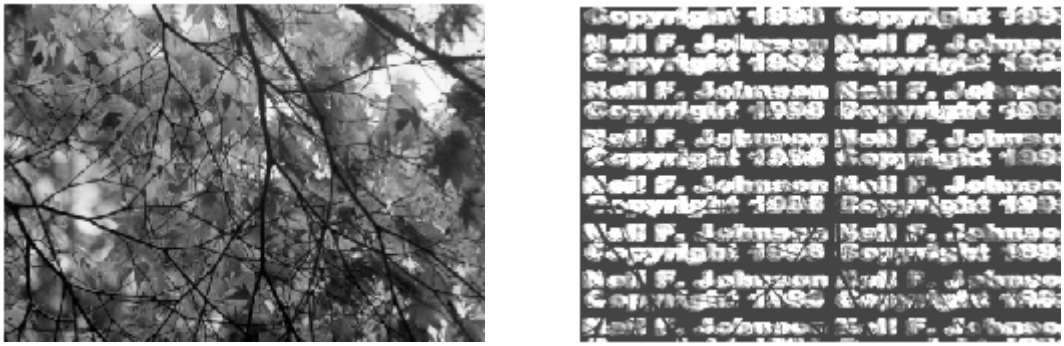
Sayısal filigranlar ikiye ayrılır. Bunlar;

1. Görünür filigran (visible watermark)
2. Görünmez filigran (invisible watermark)

Görünen filigranlar insan gözünün rahatlıkla görebileceği izlerdir. Örneğin paralar da bulunan ışığa tutunca görünen resimler (TL. deki Atatürk resmi gibi), bir başka örnek ise televizyon kanallarında o görüntünün hangi kanal yada ajans tarafından çekildiğini gösteren ekranın köşesinde bulunan bir logodur. Bir görünür filigrana saldırı, ancak o kısmın kesilerek çıkarılmasıyla yapılabilir.

Görünmeyen filigrana bir örnek ise; pasaportlarda bulunan kişiye ait seri numarasının fotoğrafın içerisine de gömülmesidir. Herhangi biri elde ettiği bir pasaporta kendi resmini yapıştırdığı zaman özel tarayıcılarla fotoğrafı tarandığında seri numarasının tutmadığı ya da olmadığı gözükecektir.

Görünmeyen filigranların görünen filigranlara göre bazı avantajları vardır. Filigran yerleri belli değildir ya da filigran olup olmadığı fark edilmeyebilir. Filigranı tüm resim içine dağıtmak genel bir uygulamadır (Şekil 1.2).



**Şekil 1.2.** Resmin tamamına filigran gömülmüş örnek.

Bu durum resmi kesme saldırılarına (cropping attacks) karşı biraz olsun koruma sağlar. Fakat dosya içerisine gömülecek olan bilgi ne kadar az ise saldırılara karşı o kadar güçlü ve güvenli olur. Bu dosya içerisindeki tekrarlılığın (redundancy) azalması için gereklidir [Johnson vd., 2000].

### 1.2.4. Steganografi (Steganography)

Steganografi bilgi gizleme yöntemlerinin en önemli alt dalıdır [Petitcolas vd., 1999]. Bu yaklaşım, bir nesnenin içerisine bir verinin gizlenmesi olarak tanımlanabilir. Dilbilim ve teknik steganografi olarak ikiye ayrılmaktadır. Bu yaklaşımla ses, sayısal resim, video görüntüleri üzerine veri saklanabilir. Görüntü dosyaları içerisine saklanacak veriler metin dosyası olabileceği gibi, herhangi bir görüntü içerisine gizlenmiş başka bir görüntü dosyası da olabilir. Steganografi konusu 2. bölümde daha ayrıntılı olarak anlatılmaktadır.

# 2. STEGANOGRAFİ

## 2.1. Steganografi Nedir?

Steganografi eski bir bilgi gizleme sanatıdır [Petitcolas vd., 1999]. Steganografi kelimesi kökleri "*στεγανος*" ve "*γραΦειν*"'den gelen Yunan alfabesinden türetilmiştir. Tam olarak anlamı "*kaplanmış yazı*" (*covered writing*) demektir [Murray ve Burchfiled, 1933].

Steganografi'nin amacı gizli mesaj ya da bilginin varlığını saklamaktır. Taşınmak istenen mesaj bir başka masum görünüşlü ortamda saklanarak, üçüncü şahısların iletilen mesajın varlığından haberdar olması engellenir.

Metin, ses, sayısal resim, video dosyaları üzerine veri saklanabilir. Bu veriler metin dosyası olabileceği gibi, herhangi bir görüntü içerisine başka bir görüntüyü gizlemekte olasıdır. Yine aynı şekilde bir ses dosyasının içine bir metin dosyası da saklanabilmektedir [Memon ve Wong, 1998] [Wang ve Wang, 2004].

Steganografi gizli bir iletişim sağlamaktadır. Amacı iki kişi arasındaki iletişimin bir üçüncü şahıs tarafından fark edilememesidir. Bilimsel ortamda Steganografi çalışmaları 1983 yılında Simmons tarafından "Prisoner Problem"'in [Simmons, 1984] tanımlanması ile başlamaktadır. Bu problemde Alice ve Bob hapishanededir ve hapishaneden kaçmak için planlar yapmaktadırlar. Fakat bu planların gardiyan Willie'ye fark ettirilmeden yapılması gerekmektedir. Eğer Willie bunu fark ederse kaçma planları suya düşecektir. Bu nedenle de çeşitli gizli haberleşme yöntemleri geliştirilmesi gerekmektedir.

Bu yaklaşımda içine bilgi gizlenen ortama örtü verisi (cover-data) veya örtü nesnesi (cover-object), oluşan ortama da stego-metin (stego-text) veya stego-nesnesi (stego-object) denmektedir [Kharrazi vd., 2004].

| Gizlenecek Mesaj + Örtü-nesnesi = Stego-nesnesi |
|---|

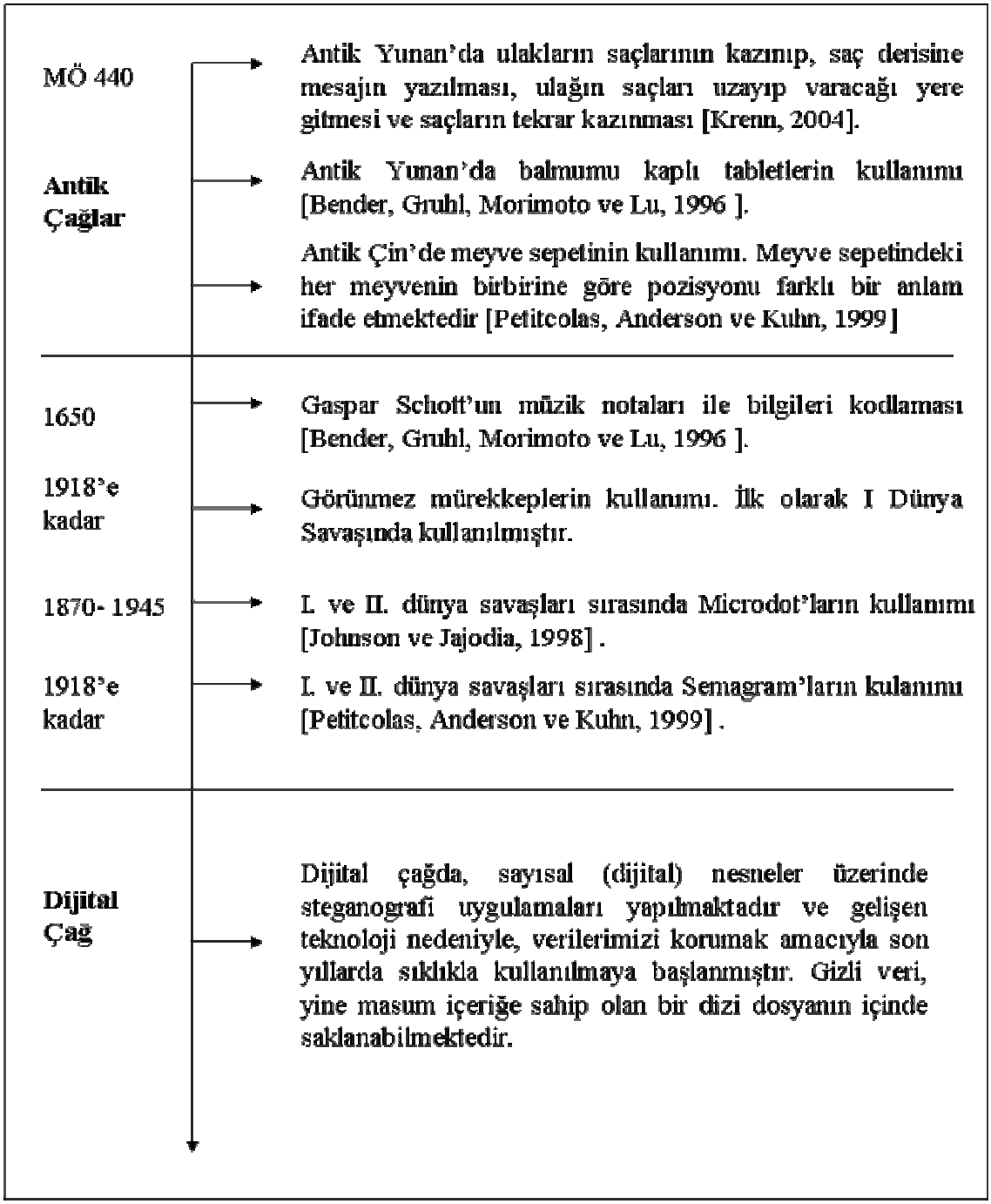
Steganografi şifrelemeye yakın olmasına rağmen şifrelemeden faklıdır. Şifreleme mesajın içeriğinin korunması ile ilgilenirken steganografi mesajın varlığının gizlenmesi ile ilgilenmektedir. Dolayısıyla steganografi bir şifreleme yöntemi değil şifrelemeyi tamamlayıcı bir öğedir [Anderson ve Petitcolas, 1998].

Steganografi birçok alanda ve çeşitli amaçlar için kullanılmaktadır. Bunlar şöyle belirtilebilir [Petitcolas vd., 1999] [Bender vd., 1996]:

- **Askeri:** Askeri durumlarda iletişimin şifrelenmesi her zaman yeterli olmamaktadır. Şifrelenmiş bir bilginin gönderildiği düşman tarafından fark edilebilir. Buna karşılık iletişim steganografik yöntemlerle yapılırsa çok daha başarılı olacaktır. Bu nedenle şifrelemeye alternatif olarak kullanılabilmektedir.
- **Filigran ve parmak izi:** Hem filigranlar hem de parmak izleri gizlendiğinde bunun belli olmamasına ve güvenli olmasına ihtiyaç duyarlar. Bu nedenle, bu işlemde steganografiden yararlanılmaktadır.
- **Sağlık alanı:** Bazı sağlık sistemleri görüntüleri ve bunlar hakkındaki açıklamaları bir yere gönderebilir ve orada saklayabilir. Bazen bu görüntüler ve açıklamalar birbirinden ayrılabilir ve bu da tehlikeli sonuçlara yol açabilir. Eğer açıklama bilgileri görüntünün içine saklanırsa böyle bir durum ortadan kaldırılır. Bu işlem için de steganografi kullanılmaktadır.

## 2.2. Steganografinin Tarihçesi

Steganografi, Antik yunan ve Heredot zamanına kadar uzanan oldukça eski bir veri gizleme yöntemidir. Steganografinin tarihsel gelişimi Şekil 2.1'de gösterilmektedir.




**Şekil 2.1.** Steganografinin tarihsel gelişimi

Yunan tarihçi Heredot, eserinde [Heredotus, M.Ö. 430], İran'da bulunan casusun, Pers istilasını Yunanistan'a nasıl ilettiğini kaydetmektedir. Yazıya göre, casus kölesinin saçını kazıtmış; istila uyarısını da kafa derisine kazıtmıştır. Daha sonra yapılacak olan, kölenin saçının yazıyı kapatacak kadar uzamasını beklemek ve bu köleyi Yunanistan'a göndermektir. Kölenin bilmesi gereken tek bilgi *"kafamı kazıyın"* olacaktır. Yine aynı çağda avcı kılığındaki bir ulağın, avladığı hayvanın karnına parşömen saklayarak Yunanistan'a girmesi anlatılmaktadır [Newman, 1940].

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

Antik dönemdeki bu basit uygulamalar steganografinin gizli iletişimdeki kullanımının insanlık kadar eski olduğunu bizlere göstermektedir [Tacticus, 1990].

Steganografi hakkında yazılan ilk kitap Johannes Trithemus (1462–1516) tarafından yazılmış olan *Steganographiæ* isimli kitaptır. 1600'lü yıllarda yaşamış olan Gaspar Schott (1608–1666) tarafından yazılmış olan *Schola Steganographica* [Schott, 1665] isimli kitapta ise müzik notalarının bilgi gizlemek için nasıl kullanıldığı anlatılmıştır. Bu yöntem birçok bilgi gizleme yöntemine de temel oluşturmuştur.

Daha sonraki yıllarda steganografi, görünmez mürekkep, metin belgelerindeki harf frekanslarını kullanma, I. ve II. Dünya Savaşlarında kullanılan mors kodları gibi uygulamalarla karşımıza çıkmaktadır [Katzenbeisser ve Petitcolas, 2000].

Ancak, çarpıcı kullanımı ikinci dünya savaşında kendini göstermektedir. İkinci dünya savaşı esnasında, Alman casusların gizli bilgileri kimyevi bir madde ile beyaz bir mendile yazdıkları ortaya çıkartılmıştır. Casus, gizli mesaj içeren bu mendili daha önce belirlenen noktalarda çöpe atmakta; alıcı ise yine kimyevi maddeler kullanarak bu yazıyı okumaktadır [Kahn, 1967].

Yine ikinci dünya savaşı döneminde Almanlar “mikrofilm” teknolojisi kullanarak “mikro noktalar” (microdot) kullanmıştırlar. Bu yöntemde A4 büyüklüğündeki herhangi bir belge veya çizim bir dizi işlem sonrasında daktilo yazısında kullanılan bir nokta kadar küçültülmektedirler. Bu yöntem kullanılarak masum içerikli bir sayfa düz metindeki i ve j harflerinin noktalarına oldukça büyük miktarda veri saklamak mümkün olmuştur [Zim, 1948].

Aşağıda, ikinci dünya savaşında kullanılan bir steganografi örneği verilmiştir [Johnson ve Jajodia, 1998].

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

Yukarıda verilen paragrafta her kelimenin ikinci harfleri yan yana getirildiğinde “*Pershing sails from NY June 1*.” Mesajı ortaya çıkmaktadır.

Günümüzde sayısal (dijital) nesneler üzerinde steganografi uygulamaları yapılmaktadır ve gelişen teknoloji nedeniyle, verilerimizi korumak amacıyla son yıllarda sıklıkla kullanılmaya başlanmıştır.

Gizli veri, yine masum içeriğe sahip olan bir dizi dosyanın içinde saklanabilmektedir. Bunlardan en ilgi çekicileri, vermiş oldukları olanaklardan dolayı, resim, ses ve video dosyalarıdır. Benzer bir şekilde düz metin dosyaları, sabit disklerdeki kullanılmayan alanlar, IP (Internet Protocol) paketlerinin ileride kullanmak üzere ayrılmış bölümleri gizli verinin saklanması için kullanılabilmektedir. Html dosyaları, exe dosyaları vb. gibi dosyalar da içlerine veri saklamada kullanılabilmektedir.

## 2.3. Steganografinin Alt Alanları

Steganografi, Dilbilim Steganografi (Linguistic Steganography) ve Teknik Steganografi (Technical Steganography) olmak üzere kendi içerisinde ikiye ayrılmaktadır [Johnson ve Rude, 2001].

Dilbilim Steganografi, taşıyıcı verinin text olduğu steganografi koludur. Burada değişiklik yapmanın çeşitli yolları vardır.

Bunlardan bazıları şöyledir:

- Grafik kullanılarak yapılabilir,
- text'in yapısı değiştirilerek yapılabilir
- ya da amacı sadece veriyi saklamak olan yeni bir text yaratılabilir.

Dilbilim Steganografi'de kullanılan yöntemler ise şunlardır:

- **Açık kodlar:** Gizli mesaj, açıkça okunabilir fakat zararsız bir mesaj haline gelir. Bu işlem; maskeleme, boş şifreler ve grid (ızgara) ile yapılmaktadır.
- **Şemagramlar:** Gizli mesaj, açık metinin ufak fakat gizli bir detayının içine gizlenmektedir. Bunun için grafiksel değişiklikler yapılmaktadır. Kullanılan yöntemler ise; farklı yazı tipleri kullanmak, eski daktilo yazılarını kullanmak, resimler içinde boşluklar kullanmak vb'dir.

Teknik Steganografi, birçok konuyu içine almaktadır. Bunları bazı başlıklar altında toplayabiliriz;

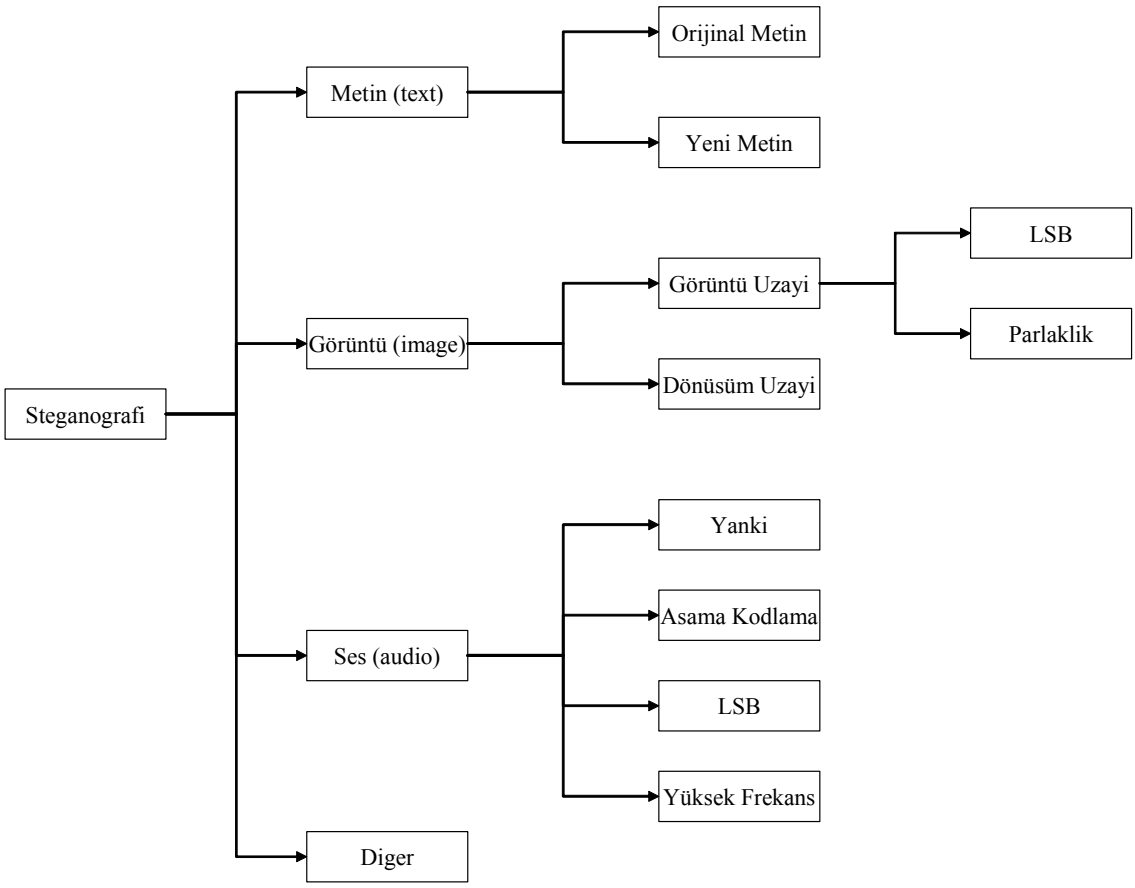
- **Görünmez mürekkep:** Geleneksel haline gelmiş olan görünmez mürekkeple yazma yöntemidir.
- **Gizli yerler:** Kimsenin göremeyeceği gizli yerlere saklama (bavul, kasa vb.)
- **Microdot'lar:** Bilgiyi noktalar halinde sayfaya gizleme
- **Bilgisayar tabanlı yöntemler:** Text, ses, görüntü, resim dosyalarını kullanarak veri gizleme yöntemleridir.

## 2.4. Steganografinin Kullanım Alanları

Sayısal steganografi kullanım alanları açısından genel olarak üçe ayrılmaktadır. Bunlar aşağıdaki gibidir:

- Metin (text) steganografi
- Görüntü (image) steganografi
- Ses (audio) steganografi

Yaygın olarak kullanılan sayısal steganografi yöntemlerinin sınıflandırılması Şekil 2.2'de verilmektedir.




**Şekil 2.2.** Sayısal Steganografi yöntemlerinin sınıflandırılması

### 2.4.1. Metin (text) Steganografi

Metin steganografi bilgi gizlenecek ortamın metin (text) olduğu steganografi koludur.

Metin steganografinin uygulanabilmesi için çeşitli yöntemler vardır.

Bunlar şu şekilde sınıflandırılabilir [Popa, 1998];

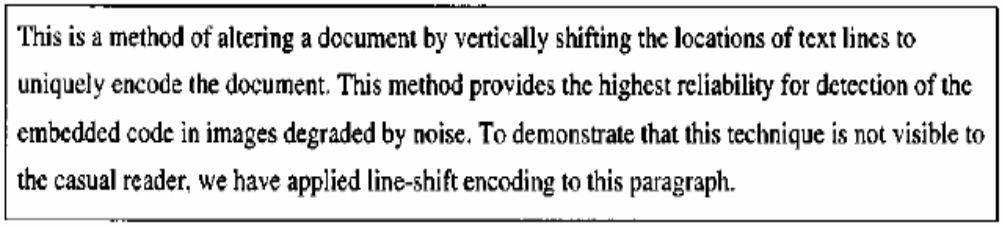
- Açık Alan Yöntemleri (Open Space Methods)
  - Satır Kaydırma Kodlaması
  - Kelime Kaydırma Kodlaması
  - Gelecek Kodlaması
- Yazımsal Yöntemler (Syntactic Methods)
- Anlamsal Yöntemler (Semantic Methods)

#### 2.4.1.1. Açık alan yöntemleri

Bu yöntemler, anormal gözükmeyen iki kelime arasında ekstra boşluklar ve satır sonu boşlukları ile çalışmaktadır. Bununla birlikte açık alan yöntemlerinin ASCII kodları ile kullanılması daha uygundur. Açık alan yöntemlerinde kullanılan kodlama yöntemleri şu şekildedir.

**Satır kaydırma kodlaması**

Bu yöntemde metin satırları düşey olarak kaydırılarak gömülecek mesajın kodlanması sağlanır. Gömülmüş kelime yine metin dosyası ya da Windows Bitmap (BMP) [Microsoft Corporation, 1990] dosya olarak açılabilir. Aşağıdaki metinde ikinci satır 1/300 inch yukarıya kaydırılmıştır. Fakat gözle anlaşılır bir fark yoktur. Bu yapılan “0” ya da “1” ile tanımlanarak kodlama işlemi gerçekleştirilir.

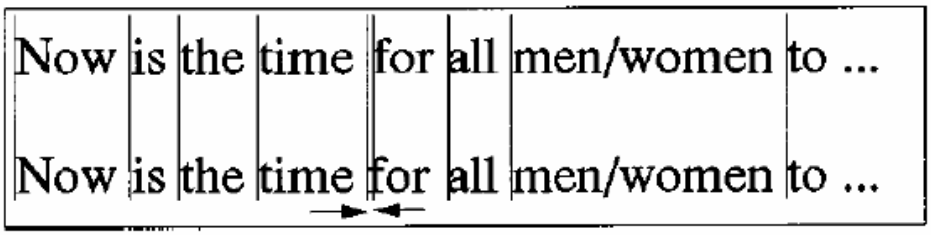

This is a method of altering a document by vertically shifting the locations of text lines to uniquely encode the document. This method provides the highest reliability for detection of the embedded code in images degraded by noise. To demonstrate that this technique is not visible to the casual reader, we have applied line-shift encoding to this paragraph.

**Şekil 2.3.** Satır kaydırma kodlaması örneği

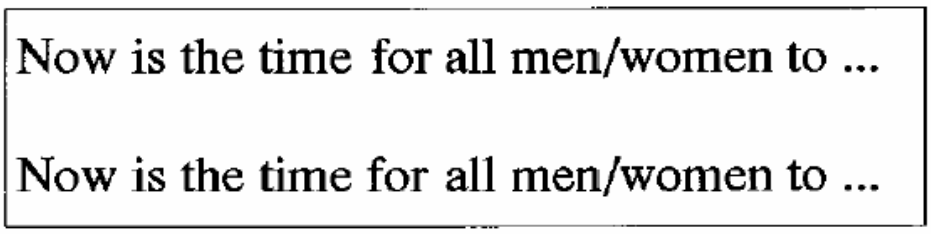
**Kelime kaydırma kodlaması**

Bu yöntemde metnin satırları yatay olarak kaydırılarak dokümanın tek olarak kodlanması sağlanır. Gömülmüş kelime yine metin dosyası ya da BMP dosyası olarak açılabilir.

Bu yöntem, dokümana uygulandığında yakın kelimeler arasında çok ta fark edilmeyen boşluklar ortaya çıkmaktadır. Bu oluşan boşluklardan dolayı dokümanın kodunun çözülmesi için eski belgeye de ihtiyaç vardır.




(a)

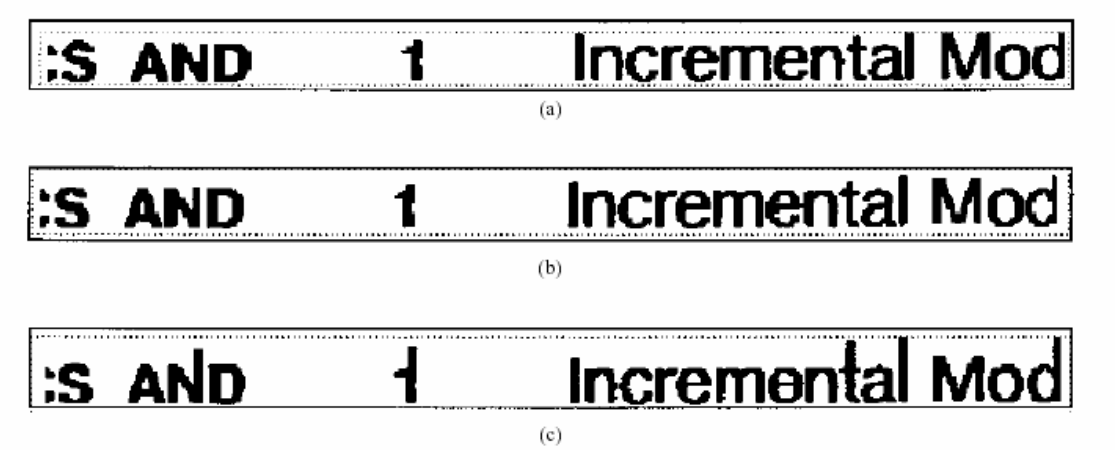



(b)

**Şekil 2.4. (a)** Üst satır'da "for" kelimesinden önce bir boşluk eklenmektedir, alt satırda for ile all arasında daha fazla boşluk vardır. **(b)** Dikey çizgiler olmadan metnin nasıl gözüktüğü

**Gelecek kodlaması**

Bu kodlama tekniği hem metin belgelerine hem de bitmap dosyalara uygulanabilmektedir. Burada kelimelerin yerleri ve bazı harflerin boylarıyla oynanmakta ve ASCII kodlarında değişiklik yapılmaktadır.




**Şekil 2.5. (a)** Herhangi bir kodlama yapılmamış orijinal metin. **(b)** Sadece seçilen karakterler üzerinde yapılmış gelecek kodlaması. **(c)** Gelecek kodlamasının abartılmış gösterimi

### 2.4.1.2. Yazımsal yöntemler

Bu yöntem, dokumanı kodlamak için noktalama işaretlerini kullanır [Bender vd., 1996]. Örneğin aşağıdaki iki cümle ilk bakışta aynıymış gibi gözükmektedir, fakat dikkatlice bakıldığında ilk cümlenin fazladan bir ‘,’ işareti içerdiği görülmektedir. Bu yapıların biri “1”, diğeri de “0” olarak belirlenmekte ve kodlama işlemi bu şekilde gerçekleştirilmektedir.

*“bread, butter, and milk”*

*“bread, butter and milk”*

#### 2.4.1.3. Anlamsal yöntemler

Bu yöntem W. Bender tarafından ortaya atılmıştır. Bu yöntemde eşanlamlı kelimelere birincil ve ikincil değerler atanmaktadır. Sonra bu değerler "1" ve "0" olarak binary'e dönüştürülmektedir.

Örneğin *"big"* kelimesi birincil, *"large"* kelimesi de ikincil olarak işaretlenmiş olsun. Birincil "1", ikincil de "0" olarak binary'e çevrilmektedir.

### 2.4.2. Görüntü (Image) Steganografi

Sayısal resimler dağıtımı en kolay ve internette hemen her sayfada karşılaşılabilecek dosyalardır. Kullanıldıkları formatlara göre farklılık göstermekle birlikte steganografi uygulamalarında en yaygın kullanılan ortamlar resim dosyalarıdır. Bu nedenle steganografi konusunda yapılan çalışmalar ve geliştirilen teknikler ağırlıklı olarak resim steganografi çerçevesinde yer almaktadır.

Görüntü dosyalarının içerisine bir metin gizlenebileceği gibi bir resim dosyasının içine bir başka resmi de gizlemek mümkündür.

Gizli bilgiyi bir resme gömme (yada gizleme) işleminde iki dosya söz konusudur. Kapak resim ya da örtü verisi (cover image) olarak adlandırılan ilk dosya, gizli bilgiyi saklayacak resim dosyasıdır. İkinci dosya ise gizlenecek bilgi olan mesajdır. Bu mesaj da stego olarak isimlendirilmektedir. Mesaj, açık metin (plain text), şifreli metin (chipher text), başka resimler veya bit dizisi içinde saklanabilecek başka bir şey olabilir. Gömme işlemi sonucunda kapak resim ve gömülü mesajın oluşturduğu dosyaya "stego resim" adı verilir.

Birçok farklı yöntem kullanılarak resimlerde bilgi gizlenebilmektedir.

Kullanılan yöntemler, gömme işlemi sırasında kullandıkları veri dikkate alınarak iki başlık altında toplanabilmektedir [Johnson ve Jajodia, 1998].

1. Uzaysal / Görüntü Alan Tekniği (Spatial / Image Domain Technique)
2. Frekans / Dönüşüm Alan Tekniği (Frequency / Transform Domain Technique)

Uzaysal Alan veya Görüntü Alan olarak adlandırılan teknik, gömme işleminde resim dosyasındaki veriyi doğrudan kullanılır. Gömme işlemin de bilgiyi gizlediği veri kümesi piksel değerlerini temsil eden kısımdır. Bu tekniğe örnek olarak yaygın olarak kullanılan En Önemsiz Bite Ekleme (Least Significant Bit Insertion - LSB) yöntemi gösterilebilir.

Frekans Alan veya Dönüşüm Alan olarak bilinen teknik ise kapak verideki değişimler üzerinde gömme işlemini uygular. Dönüşüm Alan tekniğine örnek olarak ise JPEG formatlı resim dosyalarına veri gömme işleminde kullanılan algoritmaları verebiliriz. Bu algoritmalar JPEG sıkıştırma sırasında kullanılan DCT katsayıları üzerinde veri gömme işlemini uygular.

Görüntü Steganografi ile ilgili ayrıntılı bilgiler 3. bölümde verilecektir.

### 2.4.3. Ses (Audio) Steganografi

İnsan işitme sistemi (Human auditory system-HAS) frekans aralığı yüzünden, ses sinyalleri içerisine bilgi gizleme oldukça uğraş gerektiren bir konudur. HAS 1/1.000'den daha büyük frekans aralığını fark edebilir. Aynı zamanda HAS nereden geldiği belli olmayan gürültülere de oldukça duyarlıdır.

Ses sinyalleri üzerinde uğraşırken ses dosyalarının hangi karakteristiklere sahip olduklarını bilmemiz gerekmektedir.

Ses dosyaları iki ana özelliğe sahiptirler:

- Basit nicelendirme metodu: Yüksek kaliteli sayısal seslerin 16-bit doğrusal nicelendirme ile ifadesinde en çok kullanılan yöntemdir. WAV (Windows Audio-Visual) ve AIIF (Audio Interchange File Format). Bazı sinyal bozulmaları bu formatta ortaya çıkabilir.
- Geçici seçme oranı: Ses için en çok kullanılan oranlar 8 kHz, 9.6 kHz, 10 kHz, 12 kHz, 16 kHz, 22.05 kHz ve 44.1 kHz 'dir. Bu değer frekans aralığının kullanılabilecek en üst seviyesidir.

Bir diğer sayısal gösterim ise ISO MPEG-Audio formatıdır. Bu algılama ile ilgili bir formattır. Bu yöntemde sinyal istatistiği değiştirilir. Böylece ses korunur fakat sinyal değiştirilmiş olur [Sellars, 1999].

Ses dosyalarında veri gizleme yöntemleri ise şunlardır:

- Düşük bit kodlaması (Low-bit encoding)
- Aşama kodlaması (Phase coding)
- Taft yayılması (Spread spectrum)
- Yankı veri gizlemesi (Echo data hiding)

#### 2.4.3.1. Düşük bit kodlaması

Görüntü steganografide kullanılan LSB ekleme yöntemiyle aynı şekilde gerçekleştirilir. Ses dosyasındaki verinin her baytının son bitine gizlenecek bilginin bir biti yazılır. Sonuçta oluşan değişiklik ses dosyasında gürültüye neden olmaktadır. Ayrıca dayanıksız bir yapısı vardır. Tekrar örnekleme veya kanalda oluşabilecek gürültü ile mesaj zarar görebilir veya yok edilebilir [Kim vd., 2003].

#### 2.4.3.2. Aşama kodlaması

Aşama kodlaması yöntemi de resim dosyalarında uygulanan JPEG algoritması benzeri bir yapı taşımaktadır. Gömme işleminde ses dosyası küçük segmentlere bölünür ve her segmente ait aşama (faz) gizlenecek veriye ait aşama referansı ile değiştirilir. Aşama kodlaması prosedürü aşağıdaki gibidir [Bender vd., 1996].

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

#### 2.4.3.3. Taft yayılması

Gizleme işlemini ses sinyalinin kullandığı frekans taftı üzerinde yapmaktadır. Güçlü bir yapısı olamamakla birlikte seste gürültü meydana getirmektedir [Bender vd., 1996].

#### 2.4.3.4. Yankı veri gizlemesi

Bilginin gizlenmesi taşıyıcı ses sinyali üzerine bir yankı eklenmesi ile sağlanmaktadır. Bilgi yankının gecikme miktarı, zayıflama oranı veya büyüklüğü gibi değerler kullanılarak gizlenir. İki farklı gecikme değeri kullanılarak insan kulağının algılamayacağı düzeyde 0 veya 1'in kodlanması mümkündür. Her bitin kodlanması için sinyal segmentlere bölünür. Yankı veri gizlemesi yöntemi herhangi bir gürültüye neden olmamakta veya kayıplı bir kodlama kullanmamaktadır [Gruhl vd., 1996].

### 2.4.4. Kullanılan Diğer Ortamlar

Steganografi uygulamalarında yaygın olarak kullanılan text, görüntü ve ses dosyaları haricinde, sabit disklerdeki kullanılmayan alanlar, IP (Internet Protocol) paketlerinin ileride kullanmak üzere ayrılmış bölümleri gizli verinin saklanması için kullanılabilmektedir. Yine aynı şekilde Html dosyaları, exe dosyaları vb. gibi dosyalar da içlerine veri saklamada kullanılabilmektedir.

Resim ve ses dosyalarına veri saklama yöntemleri insanın görme ve işitme sisteminin fark edemeyeceği ufak değişikliklerle bilgi gizleme mantığını temel almaktadır.

Html ve exe gibi dosyala veri saklama yöntemleri ise bu dosyaların kendi formatlarındaki esneklikleri temel alarak çalışırlar.

Örneğin html dosyalarında etiketler (tag) kullanılmaktadır. Bu etiketlerin açma ve kapama şekilleri vardır. Bir metni şekillendiren iki etiket olduğunda bunları kaparken hangisinin daha önce kapandığı hangisinin daha önce açıldığı sayfanın görünümünde fark oluşturmaz.

Örneğin:

<tr>deneme</tr> ile

<tr>deneme</tr> satırları aynı görüntüyü sağlamaktadır.

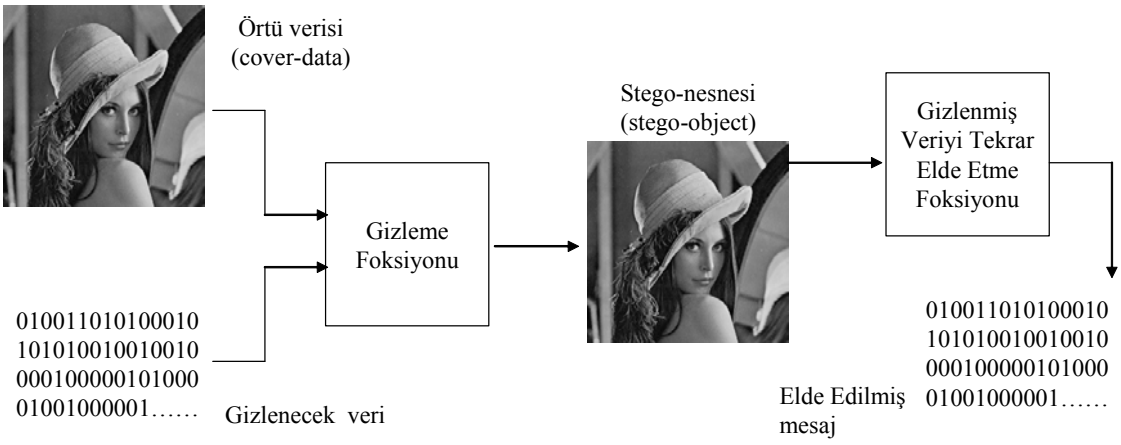
Ayrıca html dosyasında arada bırakılan boş satırların sayısı görüntüyü değiştirmemektedir. Bu durumda, bu alanlar veri saklama amacıyla kullanılabilmektedir. Ancak bu yöntemlerin saklama kapasitesi düşüktür ve steganalitik yöntemlere karşı dayanıklılığı azdır.

Exe dosyalarında da benzer mantıkla hareket edilmektedir. Komut setlerinde aynı işlevi gören farklı komutlar olabilmekte ve bunlardan hangisinin kullanıldığı oluşan exe dosyasının çalışmasında değişikliğe neden olmamaktadır. Bu durum veri saklama amacıyla kodlanabilmektedir. Ancak yine burada da saklama kapasitesindeki düşüklük gündeme gelmektedir [Atıcı, 2005].

# 3. GÖRÜNTÜ (IMAGE) STEGANOGRAFİ

Sayısal resimler dağıtımı en kolay ve internette hemen her sayfada karşılaşılabilecek dosyalardır. Kullanıldıkları formatlara göre farklılık göstermekle birlikte küçük boyutları ve içerdikleri verinin genellikle fazlalık (redundancy) içermesi sebebiyle steganografi uygulamalarında en yaygın kullanılan sayısal ortamlar resim dosyalarıdır. Bu nedenle steganografi konusunda yapılan çalışmalar ve geliştirilen teknikler ağırlıklı olarak görüntü steganografi çerçevesinde yer almaktadır.

Görüntü dosyaları için bir steganografik sistem Şekil 3.1'de gösterilmektedir. Gönderici bir gizleme fonksiyonu kullanarak bir steganogram yaratır. Gizleme fonksiyonu, verinin saklanacağı taşıyıcı ortam ve gizlenecek veri olmak üzere iki parametreye sahiptir [Westfeld ve Pfitzmann, 1999].




**Şekil 3.1.** Steganografik sistem

Herhangi bir steganografik sisteminin temelde şeffaflık (transparency) ve sağlamlık (robustness) şartlarını sağlaması gerekmektedir. Şeffaflık saklanan verinin tespit edilememesi ve fark edilememesini ifade ederken sağlamlık saklanan verinin çıkartma işleminde düzgün bir şekilde geri getirilmesini anlatmaktadır.

Görüntü steganografide, bilgilerin görüntü dosyaları içerisine saklanmasının çeşitli yöntemleri vardır.

Şekil 3.1'de Gizleme Fonksiyonu olarak adlandırılan ve bilgi gizlemede en çok kullanılan yöntemler aşağıda gösterilmiştir:

- En önemsiz bite ekleme
- Maskeleme ve filtreleme
- Algoritmalar ve dönüşümler [Sellars, 1999].

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

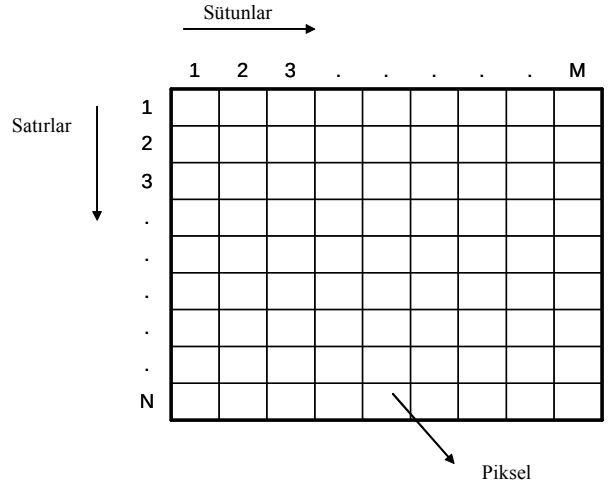
Maskeleme ve filtreleme yöntemleri genellikle 24 bit resimler için kullanılmakta olup resmin en önemsiz alanlarının tespit edilerek buralarda saklama yapılmasını temel almaktadır. Bu yöntemler genelde filigran uygulamalarında karşımıza çıkmaktadır. Maskeleme teknikleri JPEG formatındaki resim dosyaları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

Dönüşümler ise yine daha çok JPEG dosyalar üzerinde kullanılmaktadır. En yaygın olarak kullanılan dönüşümler ise DCT ve DFT'dir.

## 3.1. Sayısal Resmin Yapısı

Sayısal (dijital) resim $N$ satır ve $M$ sütunluk bir dizi ile temsil edilir. Genellikle satır ve sütun indeksleri $y$ ve $x$ veya $r$ ve $c$ olarak gösterilebilir.

Bir resim dizisinin elemanlarına piksel denir. En basit durumda pikseller 0 veya 1 değerini alırlar. Bu piksellerden oluşan resimlere ikili (binary) resim denir.




**Şekil 3.2.** Sayısal resmin temel yapısı

1 ve 0 değerleri sırasıyla aydınlık ve karanlık bölgeleri veya nesne ve zemini (nesnenin önünde veya üzerinde bulunduğu çevre zemini) temsil ederler [Tunçkanat ve Sağıroğlu, 2002]. Sayısal (dijital) görüntü dosyaları renkli olarak genellikle 8 yada 24 bit; gri-seviye görüntüler 1-2-4-6 yada 8 bit olabilirler.

Gri-seviye resimlerde her piksel, 0 (siyah) ile 255 (beyaz) arasında tam sayı değer alabilen 1 byte ile temsil edilmektedir. 0–255 arasındaki değerler gri'dir ve bundan dolayı bir resme ait tam sayı "*gri ton seviye*" (gray level) olarak isimlendirilmektedir [Potdar ve Chang, 2004].

8 bitlik renkli görüntülerde piksel başına 1 byte kullanılır. 8 bitlik görüntüler renk sınırlaması yüzünden çok iyi bir sonuç vermemektedir. Saklanacak bilgi, saklama ortamını çok fazla değiştirmeyecek şekilde dikkatlice seçilmelidir. Orijinal görüntüde son bite ekleme işlemi yapıldığında, renk girişi göstergeleri değişmektedir. 8 bitlik görüntülerde 4 basit renk (WRBG) kullanılmaktadır. Bunlar; beyaz (White-W), kırmızı (Red-R), mavi (Blue-B) ve yeşildir (Green-G).

Bu renklerin renk paletinde karşılık gelen girişleri ise sırasıyla 0 (00), 1 (01), 2 (10), 3 (11) şeklindedir [Johnson ve Jajodia, 1998].

24 bit resimler ise bir piksel başına 3 byte kullanmaktadır. Her pikselin rengi; Kırmızı (red), Yeşil (green), Mavi (blue) olmak üzere üç ana renkten elde edilmektedir. Buna pikselin RGB değeri denmektedir [Morkel vd., 2005].

## 3.2. Resim Dosyalarının Sıkıştırılması

Resim dosyalarında "*kayıplı (lossy)*" ve "*kayıpsız (lossless)*" olarak iki sıkıştırma yöntemi vardır. Her iki yöntem de dosya büyüklüğünü azaltmaktadır. Ancak gömülü bilginin etkilenmesi açısından farklı sonuçlar vermektedirler [Moerland, 2003].

Kayıpsız sıkıştırma, orijinal mesajın doğru olarak elde edilmesini sağlamaktadır. Bu nedenle orijinal bilginin eksiksiz elde edilmesi gereken durumlarda tercih edilmektedir. GIF (Graphic Interchange Format) [CompuServe, 1990] ve 8 bit BMP (Windows Bitmap) formatları bu yapıdadır.

Kayıplı sıkıştırma, dosya büyüklüğünü büyük ölçüde azaltmakta ancak resmin bütünlüğünü korumamaktadır. JPEG resimler bu tip sıkıştırma kullanan resimlerdir. Sıkıştırma sonrasında gizlenen bilgide kayıplar meydana gelebilmektedir. Kullanılan kayıplı sıkıştırma algoritmasına bağlı olarak JPEG formatı [Wallace, 1991], yüksek kaliteli sayısal resimlere yakın sonuç vermektedir.

BMP formatındaki resimler içinse herhangi bir sıkıştırma işlemi uygulanmamaktadır. Bu nedenle veri kaybı olmamaktadır.

## 3.3. Veri Gömme İşlemi

Gizli bilgiyi bir resme gömme işlemin iki dosya söz konusudur. Kapak resim (cover image) olarak adlandırılan ilk dosya, gizli bilgiyi saklayacak resim dosyasıdır. İkinci dosya ise gizlenecek bilgi olan mesajdır. Mesaj açık metin (plain text), şifreli metin (chipher text), başka resimler veya bit dizisi içinde saklanabilecek başka bir şey

olabilir. Gömme işlemi sonucunda kapak resim ve gömülü mesajın oluşturduğu dosyaya "stego resim" adı verilmektedir.

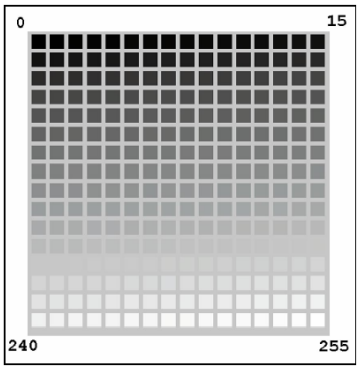
Birçok steganografi yazılımı JPEG formatını desteklemez veya kullanımını tavsiye etmezken, 24 bit BMP resimlerin kullanılmasını tercih etmektedir. Diğer alternatifler ise gri tonlamalı ve 256 renk resimlerdir. Söz edilen 256 renk resimlerin en yaygın kullanılanı GIF formatıdır.

GIF formatlı resim dosyaları ve 8 bit BMP resimlerde her piksel bir byte ile gösterilir. Bu tip resimler resimde kullanılan renkleri içeren 256 renkli bir palet taşırlar. Her piksel bu palette bir renge karşılık gelen 1 byte'lık değeri taşımaktadır (Şekil 3.3).




**Şekil 3.3.** 256 renk resim ve kullandığı palet

Birçok steganografi uzmanı da 256 gri tonlamalı resimlerin kullanımını önermektedir [Aura, 1995]. Gri tonlamalı resimlerin tercih edilme sebebi, koyuluğun her değer için çok küçük farklarla artmasıdır. Bu tip resimler de palet içermektedir (Şekil 3.4) ve palet değerlerindeki küçük değişimler gözün fark edemeyeceği kadar azdır. Bazı gri tonlamalı resimler 4 bitliktir ve 16 farklı gri ton içermektedir. Bu yapıdaki resimlerde değişimler daha belirgin olmaktadır.




**Şekil 3.4.** 256 gri tonlamalı resme ait palet

Gri tonlamalı resimler, steganografi için en iyi sonucu verdiğine göre ince renk değişimlerini çok miktarda içeren resimler de oldukça etkilidir. Düz renkli büyük kısımlar içeren bir resim ise iyi bir seçenek değildir. Bu düz renkli kısımlarda gömülü mesajın oluşturacağı değişimler dikkat çekici olabilir.

## 3.4. Veri Gömme Yöntemleri

Birçok farklı yöntem kullanılarak resimlerde bilgi gizlenebilir. Kullanılan yöntemleri, gömme işlemi sırasında kullandıkları veriyi dikkate alarak iki başlık altında toplayabiliriz [Johnson ve Jajodia, 1998].

1. Uzaysal / Görüntü Alan Tekniği (Spatial / Image Domain Technique)
2. Frekans / Dönüşüm Alan Tekniği (Frequency / Transform Domain Technique)

Uzaysal Alan veya Görüntü Alan olarak adlandırılan teknik, gömme işleminde resim dosyasındaki veriyi doğrudan kullanılır. Gömme işlemin de bilgiyi gizlediği veri kümesi piksel değerlerini temsil eden kısımdır. Bu tekniğe örnek olarak ileride değinilecek olan ve yaygın olarak kullanılan En Önemsiz Bite Ekleme (Least Significant Bit Insertion - LSB) yöntemini gösterebiliriz.

Frekans Alan veya Dönüşüm Alan olarak bilinen teknik ise kapak verideki değişimler üzerinde gömme işlemini uygular. Dönüşüm Alan tekniğine örnek olarak ise JPEG formatlı resim dosyalarına veri gömme işleminde kullanılan algoritmaları verebiliriz. Bu algoritmalar JPEG sıkıştırma sırasında kullanılan DCT katsayıları üzerinde veri gömme işlemini uygular [Johnson ve Jajodia, 1998].

## 3.5. Görüntü Dosyalarında Steganografik Yöntemler

Görüntü Dosyalarında bilgiyi gizlemek için kullanılan çeşitli yöntemler vardır. Bu yöntemler 3 gruba ayrılabilir [Erkin ve Örencik, 2005];

**Değiştirmeye dayalı yöntemler:** Bu grup yöntemlere LSB Yöntemi ya da Amplitude (Bolluk) Modülasyonu kullanılarak bilgi gizleme verilebilir. Burada bilgi gizlemek için renk değerleriyle oynanabilir ya da palet değiştirilebilir.

Renk değerleriyle oynama en basit yöntemdir. Renk değerlerinin düşük anlamlı bitleri ile gizli verinin bileri değiştirilir. Değişim, insan gözü tarafından algılanmaz. Gizli veri, "*gürültü (noise)*" olarak resme eklenir. Bu yöntem yüksek oranda veri gömme şansı verir, fakat resim üzerinde yapılacak değişimlere karşı oldukça hassastır.

Palet ile oynamada ise renk bilgilerinin palet üzerinde tutulduğu resim dosyaları kullanılır. Paletteki sıralama değiştirilir. Bunun sonucunda resim bozulabilir. Ayrıca resim türü değiştirildiğinde tüm yapılanlar yok edilir

**İşaret işlemeye dayalı yöntemler**: Bu yöntemler çeşitli dönüşümlerin kullanıldığı yöntemlerdir. DCT, DFT gibi dönüşümler kullanılabilir.

İşaret işlemeye dayalı yöntemler resim üzerinde yapılan değişikliklere karşı da dayanıklıdırlar fakat resmi bozabilirler. Resimde bozulma olup olmayacağı da ancak işlem sonrası anlaşılabilir. Ayrıca, her 64 adetlik bloklara 1 bit gömülebilmesi, saklanabilecek veri miktarını önemli oranda düşürmektedir.

**Spektrum yayılmasına dayalı yöntemler:** Tayf (Spektrum) yayılmasına dayalı yöntemler de son yıllarda oldukça fazla kullanılmaya başlanmıştır. Tayf yayılması askeri iletişimde oldukça yoğun kullanılmaktadır.

Bu yöntemde gönderilmek istenen mesaj ihtiyaç duyduğu frekans bandından çok daha fazlasına dağıtılır. Üçüncü bir kişi araya girip bir ya da birden fazla frekans bandında bozulmalara neden olsa bile, alıcı geri kalan frekans bantlarındaki bilgiler ile asıl mesajı elde edebilmektedir. Gizli mesaj birden fazla bantta yayılarak resme gürültü olarak eklenebilir.

Yukarıdaki yöntemleri kullanarak bilgi gizleyen steganografik algoritmalardan en yaygın olarak kullanılanları şunlardır;

- Patchwork Algoritması
- Amplitude (Bolluk) Modülasyonu kullanılarak bilgi gizleme
- Superposition Algoritması
- SSIS (Spread Spectrum Image Steganography) Yöntemi
- Frekans Domaini İçine Veri Saklanması
- Son Bite Ekleme (LSB-Least Significant Bit Insertion) yöntemi

### 3.5.1. Patchwork Algoritması

Bender tarafından ortaya atılan ve halen sıklıkla kullanılan algoritmadır. Bu algoritma, bilgiyi Gauss dağılımı gösteren bir istatistiğe sahip örtü verisinin içine gizlemeyi amaçlayan bir istatistiksel yönteme dayanır [Bender vd., 1996] [Gruhl vd., 1996]. Bu algoritma genelde filigran (watermarking) uygulamalarında kullanılmaktadır.

Bu algoritma genellikle 256 bit gri-seviye (gray-scale) için kullanılmaktadır. Resim içinde rasgele olarak iki nokta seçilir ( $A$ ve $B$ ). $a_i$ , $A$ noktasının parlaklığı, $b_i$ ise $B$ noktasının parlaklığıdır. Bir resmi şifrelemek için 4 adımda işlem yapılır.

**Adım 1:** $(a_i, b_i)$ çiftini seçmek için sahte rasgele (pseudo random) numara üreteci ile bir belirli anahtarın kullanılmasını gerektirir.

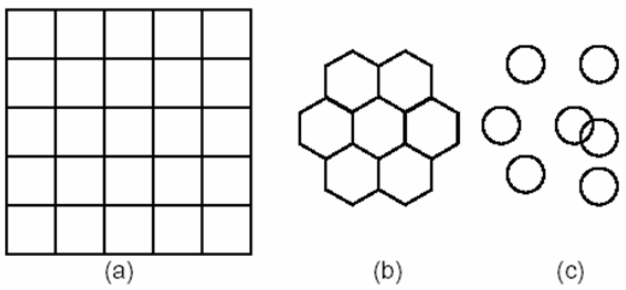
**Adım 2:** $a$ parçasının parlaklığı $\delta$ [1] kadar arttırılır.

**Adım 3:** $b_i$ parçasının parlaklığı ise aynı $\delta$ kadar azaltılır.

**Adım 4:** Son adımda ilk üç adım $n$ için tekrarlanır (n'in değeri 10.000 civarındadır)

Burada yama (patch) şekilleri de oldukça önem kazanmaktadır. Olası üç adet tek boyutlu yama şekli vardır. Eğer keskin kenar içeren küçük yamalar seçersek, yamanın enerjisi görüntü analizinin yüksek frekanslı kısmı içerisinde yoğunlaşacaktır. Fark edilmesi oldukça zordur fakat filtreleme sonucunda kolaylıkla elde edilebilir. Diğer bir olasılık yumuşatılmış kenarlar içeren yamalar kullanılmasıdır. Bu durumda bilgiler düşük-frekans analizi içinde kalacaktır. Üçüncü olasılık ise ilk iki olasılığın birleştirilmesidir. Bu şekilde yamanın enerjisi dağıtılmaktadır.

Şekil 3.5. (a)'da doğrusal bir kafes biçimi kullanılmıştır. Fakat bu pek tercih edilen bir yöntem değildir. Şekil 3.5. (b) a'ya alternatif olarak gösterilmiştir. Fakat en tercih edilir çözüm Şekil 3.5. (c)'de verilmektedir. Buradaki yamalar rasgele olarak dağılmakta ve seçilebilmektedir. Akıllıca bir dağılımla her türlü çözünürlükte iyi sonuç verebilecektir.




**Şekil 3.5.** Yama çeşitleri

[1] $\delta$ değeri, 256'lık gri-seviye renk aralığı içerisinde 1 ile 5 arasında bir değerdir

### 3.5.2. Amplitude (Bolluk) Modülasyonu Kullanılarak Bilgi Gizleme

Amplitude modülasyonu kullanılarak işaret bitleri mavi kanaldaki piksel değerleri değiştirilerek gömülür. Bu değişimler ışığın oranına ve bitin değerine bağlı olarak arttırma ya da eksiltme yoluyla yapılmaktadır [Kutter vd., 1997].

$s$ ; saklanacak tek bir bit;

$I = \{R, B, G\}$ görüntü;

$p(i, j)$ ; $I$ görüntüsü içerisinde sahte rasgele (pseudorandom) olarak seçilmiş bir pozisyon;

$K$ 'da üretilen anahtar olarak tanımlanmaktadır.

Gizlenecek bit $L = 0.299R + 0.587G + 0.114B$ olmak üzere B Mavi kanal (blue channel)'ın değiştirilmesiyle p pozisyonuna saklanır.

$$B_{ij} = B_{ij} + (2s - 1)L_{ij}q \tag{3.1}$$

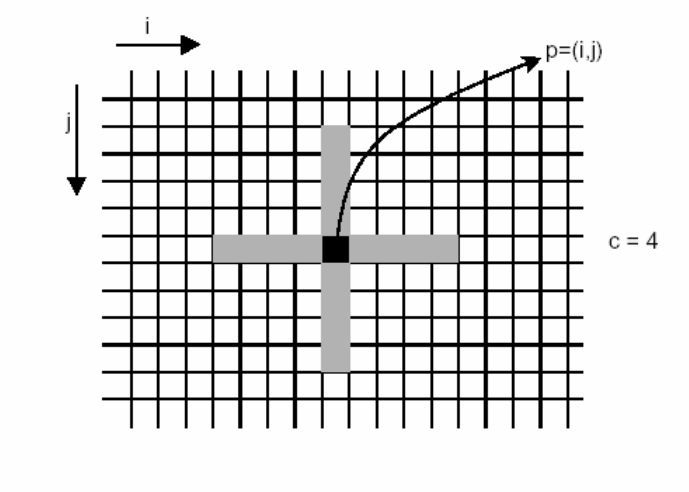
Buradaki $q$, imzanın gücüne göre belirlenmiş olan bir sabittir. $q$ 'nun değeri, veri saklamanın amacına bağlı olarak seçilmektedir. $q$ 'nun değeri değiştirilerek veri gizleme (data hiding) yada doküman işaretleme (document marking) işlemleri yapılabilmektedir.

Gizlenen bitleri sırasıyla geri getirmek için, örtü verisinin (cover data) ilk renk değerlerinin tahmin edilmesi gerekmektedir. Bu tahmin, $p(i, j)$ 'nin komşularının mavi kanal $(B)$ değerlerinin lineer kombinasyonuna dayanmaktadır. En iyi performansın çapraz komşular kullanılarak elde edileceği düşünülmektedir. Tahmini değer şu şekilde hesaplanmaktadır.

$$\hat{B}_{ij} = \frac{1}{4c}\left( \sum_{k=-c}^{c} B_{i+kj} + \sum_{k=-c}^{c} B_{ij+k} - 2B_{ij} \right) \tag{3.2}$$

Burada c değeri, çapraz şekilde olan komşuların uzunluğudur. Gizlenen bitleri geri getirmek için Tahmini değer ile pikselin şu anki değeri arasındaki fark alınır.

$$\delta = B_{ij} - \hat{B}_{ij} \tag{3.3}$$




**Şekil 3.6.** Çapraz (cross-shapes) şekilde alınan komşular

### 3.5.3. Superposition Algoritması

Superposition algoritması [Pitas ve Nikolaidis, 1996] daha çok filigran uygulamaları için kullanılmaktadır. $NxM$ çözünürlüğe sahip bir görüntü şu şekilde ifade edilmektedir.

$$I = \{x_{nm}, n \in \{0,.....,N-1\}, m \in \{0,.....,M-1\}\} \tag{3.4}$$

Buradaki $X_{nm} \in \{0,1,.....,L-1\}$, $(n,m)$ pikselinin koyuluk seviyesini belirlemektedir. Gizlenecek bitler, 1 ya da 0 değeri alabilen aynı büyüklükteki ikili (binary) çiftler olarak gösterilebilir.

$$S = \{s_{mn}, n \in \{0,....,N-1\}, m \in \{0,....,M-1\}, s_{nm} \in \{0,1\}\} \tag{3.5}$$

Bu aşamadan sonra $I$, $S$ kullanılarak iki eş büyüklükte alt kümeye bölünebilir.

$$
\begin{aligned}
&A = \{x_{nm} \in I, s_{nm} = 1\} \\
&B = \{x_{nm} \in I, s_{nm} = 0\} \\
&|A| = |B| = \frac{|I|}{2} = \frac{N \times M}{2} = P \\
&I = A \cup B
\end{aligned}
\tag{3.6}
$$

Filigran (işaret), görüntü üzerinde şu şekilde gibi ilave edilir. $C = \{x_{nm} \otimes k, x_{nm} \in A\}$. Buradaki $\otimes$ işlemi superposition kuralı (Einstein, 1905) olarak bilinmektedir. İşaretlenmiş görüntü şu şekilde verilmektedir.

$$I_s = C \cup B \tag{3.7}$$

### 3.5.4. SSIS (Spread Spectrum Image Steganography) Yöntemi

SSIS yöntemi, sayısal görüntünün içindeki bilgi bitlerinin işaretleme kalitesini, saklama ve geri getirme işlemlerinin gözlemleyen biri tarafından fark edilemeyecek şekilde yapılması yeteneği sağlar [Marvel vd., 1998]. Gizli bitlerin elde edilmesi için orijinal görüntüye ihtiyacı yoktur.

SSIS yöntemi tayf (spectrum) yayılım iletişimi, hata kontrol kodlaması, görüntü işleme gibi tekniklerle birleştirilebilmektedir. Bu düşünce tarzıyla saklanacak bilgi bir gürültü (noise) içine konularak görüntü içerisine yerleştirilir. Gürültünün düşük güçte olmasından ve kod-çözme işleminden dolayı mükemmel değildir.

**Adım 1:** Mesaj seçime bağlı olarak *key*1 ile şifrelenmekte

**Adım 2:** Sonra düşük oranlı hata kontrol kodu (error control code-ecc) ile kodlanarak, $m$ kodlanmış mesajı üretilmektedir.

**Adım 3:** Gönderici; aynı zamanda $n$ sıralamasına sahip bir yayılım üretmek için *key*2 'yi sağlar.

**Adım 4:** Daha sonra bu iki değer ($m$ ve $n$) bir modülasyon işlemine girer ve $s$ gömülü sinyali elde edilir.

**Adım 5:** Bu gömülü sinyal bir *key*3 ile birleştirme (interleaving) işlemine tabi tutulur.

**Adım 6:** Daha sonra örtü verimiz (cover data) olan $f$ ile işleme girerler.

**Adım 7:** Sinyal orijinal örtü verimizin ( $f$ ) içine yerleştirilerek stego-image elde edilir.

**Adım 8:** Son olarak ta bu şekilde alıcıya gönderilir.

### 3.5.5. Frekans Domaini İçine Veri Saklanması

Yine filigran (watermark) teknolojileri için kullanılan bu yöntem görüntülerin dönüştürülmesi (transform) temeline dayanmaktadır. Görüntü dönüştürülmesi için genellikle ayrık kosinüs dönüşümü (Discrete Cosine Transform- DCT) kullanılmaktadır. Bunun dışında kullanılan dönüşüm algoritmaları ise; Ayrık Fourier Dönüşümü (DFT), Walsh dönüşümü ya da Wavelet (dalga) dönüşümüdür Aşağıda algoritmanın çalışması gösterilmektedir [Barni vd., 1997].

$M$ filigranın büyüklüğü olmak üzere; $X = \{x_1, x_2, ....., x_M\}$ olarak verilmektedir. Bu teknik görüntü üzerinde bazı DCT katsayıları hesaplanarak bilgiye eklenip gizlenmesi için kullanılır.

Öncelikle $MxN$ piksellik $I$ görüntüsünün DCT'si hesaplanır. Sonra $L + M$ DCT katsayısı seçilerek $T$ vektörü üretilir.

$$T = \{t_1, t_2, ....., t_{L+M}\} \tag{3.8}$$

Daha sonra aşağıdaki formül kullanılarak yeni bir $T$ vektörü üretilir.

$$t_1^{'} = \begin{cases} t_i, & i = 1, ...., L \\ t_i + \alpha t_i x_i, & i = L+1, ....., L+M \end{cases} \tag{3.9}$$
$$T' = \{t_1', t_2', ....., t_{L+M}'\}$$

### 3.5.6. Son Bite Ekleme (Least Significant Bit Insertion-LSB) Yöntemi

En önemsiz bite ekleme yöntemi (Least Significant Bit Insertion Methods) [Cox vd., 1996] yaygın olarak kullanılan ve uygulaması basit bir yöntemdir. Fakat yöntemin dikkatsizce uygulanması durumunda veri kayıpları ortaya çıkmaktadır.

Bu yöntemde; resmi oluşturan her pikselin her byte'nın en önemsiz biti olan son biti değiştirilerek o bitin yerine gizlenmesini istediğimiz verinin bitleri sırasıyla verinin başlangıcından itibaren birer birer yerleştirilmektedir.

Burada her sekiz bitin en fazla bir biti değişikliğe uğratıldığından ve eğer değişiklik olmuşsa da değişiklik yapılan bitin byte'ın en az anlamlı biti olmasından dolayı, ortaya çıkan steganogramdaki (= örtü verisi + gömülü veri) değişimler insan tarafından algılanamaz boyutta olmaktadır.

Resimlerin özelliklerine göre son bite ekleme yönteminin çalışma şekilleri aşağıda verilmiştir.

#### 3.5.6.1. Gri seviye resimler üzerinde LSB yönteminin uygulanması

Gri-seviye resimlerde her piksel, 0 (siyah) ile 255 (beyaz) arasında tam sayı değer alabilen 1 byte ile temsil edilmektedir. 0-255 arasındaki değerler gri'dir ve bundan dolayı bir resme ait tam sayı "*gri ton seviye*" (gray level) olarak isimlendirilmektedir [Alwan vd., 2005].

| | Renk değeri | İkilik Sistemdeki Karşılığı | Rengi |
|---|---|---|---|
| Orijinal piksel | 182 | 10110110 | |
| Bilgi saklanmış piksel | 183 | 1011011**1** | |

Örneğin, renk değeri 182 olan bir pikselin içine ikilik sayı sistemindeki 1 değeri saklandığında oluşan piksel ve renk değeri yukarıda gösterilmektedir.

Yukarıdaki renklerden de görüleceği gibi iki renk arasında gözle fark edilemeyecek kadar az bir değişim vardır. Son bitin 1 ya da 0 olması gözle görülebilir bir fark yaratmamaktadır [Farid, 2003].

#### 3.5.6.2. 8-bit Renkli Resimler ve LSB yönteminin uygulanması

8 bitlik görüntülerde piksel başına 1 byte kullanılmaktadır. 8 bitlik görüntüler renk sınırlaması yüzünden çok iyi bir sonuç vermemektedir. Saklanacak bilgi, saklama ortamını çok fazla değiştirmeyecek şekilde dikkatlice seçilmelidir. Orijinal görüntüde son bite ekleme işlemi yapıldığında, renk girişi göstergeleri değişmektedir. 8 bitlik görüntülerde 4 basit renk (WRBG) kullanılmaktadır. Bunlar; beyaz (White-W), kırmızı (Red-R), mavi (Blue-B) ve yeşildir (Green-G).

Bu renklerin renk paletinde karşılık gelen girişleri ise sırasıyla 0 (00), 1 (01), 2 (10), 3 (11) şeklindedir.

Örnek olarak verilen orijinal görüntü pikselleri “Beyaz, beyaz, mavi, mavi” (00 00 10 10) ise 10 sayısının ikilik (binary) tabandaki karşılığı olan 1010 değeri bu piksellere gizlendiğinde, yapılan değişiklikler sonucunda görüntünün yeni piksel değerleri aşağıdaki gibi elde edilmektedir.

0**1** 00 1**1** 10

Bu değerler de renk paletinde sırasıyla kırmızı, beyaz, yeşil ve mavi değerlerine karşılık gelmektedir [Johnson ve Jajodia, 1998]. Piksellerin renk değerleri oldukça değiştiğinden, gözle fark edilebilecektir ve bu kabul edilemez bir durumdur. Veri-gizleme uzmanları bu nedenle 8 bitlik renkli görüntüler yerine gri-seviye görüntülerin kullanılmasını daha uygun bulmaktadırlar [Sellars, 1999].

### 3.5.6.3. 24-bit Renkli Resimler ve LSB yönteminin uygulanması:

24 bit resimler bir piksel başına 3 byte kullanmaktadır. Her pikselin rengi "Kırmızı (red), Yeşil (green), Mavi (blue)" olmak üzere üç ana renkten elde edilmektedir. Buna pikselin RGB değeri denmektedir.

Her byte'ta son biti değiştirmek suretiyle bir piksel'de 3 bitlik bilgi saklanabilir. Yani 24 bit derinliğine sahip 1024x768 piksel boyutundaki bir resim, bilgi saklamak için kullanılabilir 2.359.296 bit (294.912 byte)'e sahiptir. Gizlenmek istenen mesaj, saklama işleminden önce sıkıştırılırsa çok daha fazla sayıda bilgi resmin içine gizlenebilir.

10010101 00001101 11001001 (149,13,201)
10010110 00001111 11001010 (150,15,202)
10011111 00010000 11001011 (159,16,234)

Orijinal görüntü bitleri yukarıdaki gibi verilen 3 pikselin içine "*10110110*1" bilgisi gizlendiğinde oluşan yeni piksel değerleri aşağıdaki gibi olmaktadır.

10010101 0000110**0** 11001001 (149,12,201)
10010111 0000111**0** 11001011 (151,14,203)
10011111 00010000 11001011 (159,16,234)

Yukarıdaki örnekte sadece 4 bitte değişiklik yaparak bilgi gizlenmektedir. Bu yöntemde en az değişikliği yaparak sonuca gitmek ve gizlenecek bilgi 9 bitten az ise hangi bitlerin yok sayılacağını belirlemek oldukça önemlidir [Kessler, 2001].

### 3.5.7. Steganografik Yazılımlar

Yukarıda anlatılan yöntemleri kullanarak görüntü dosyaları içerisine veri gizleyen birçok steganografi yazılımı geliştirilmiştir. Bunlardan bazıları aşağıda anlatılmıştır.

#### 3.5.7.1. Outguess

Bu program verilerin ilgili bitlerine mesaj saklayan bir steganografi uygulamasıdır. Uygulama taşıyıcı dosyanın türünden, bu dosyanın veri saklamaya uygun bitlerini getirebilen ve değiştirilen bitleri geri yazabilen bir çekirdek program sağlandığı müddetçe bağımsız olarak çalışmaktadır. PNM ve JPEG formatlar üzerinde çalışmaktadır [Outguess].

#### 3.5.7.2. Stego Machine

Bu program her türlü veriyi JPEG ve GIF formatındaki resimlerinin içine saklayabilmektedir. İsteğe bağlı olarak veriyi saklamadan önce 3DES algoritmasıyla şifreleme imkânı da sunmaktadır. Bu program veriyi gizlemek için iki teknik kullanmaktadır. Bunlardan ilki veriyi dosyanın sonuna ekleme, diğeri ise en az öneme sahip biti değiştirme yöntemidir. Dosyanın sonuna ekleme yöntemi hem JPEG hem de GIF formatındaki dosyalara uygulanırken en önemsiz biti değiştirme yöntemi sade GIF formatındaki dosyalara uygulanmaktadır [Stego Machine].

#### 3.5.7.3. bmpSteg

Bu uygulama BMP formatındaki resimlerin içine veri gizlemektedir. Java programlama dili ile yazılmıştır. Sıkıştırılmamış BMP resimleri taşıyıcı dosya türü

olarak desteklemektedir. Veri saklama oranı 1/255 dir. Programın saklama kapasitesi düşüktür [Huitsing].

#### 3.5.7.4. Stella

Bu uygulama her türlü veriyi BMP, JPEG ve GIF formatındaki resim dosyalarının içine saklayabilmektedir. Veri saklamadan önce özel bir anahtar vasıtasıyla şifrelenmektedir [Stella].

#### 3.5.7.5. SecureEngine Professional

Bu uygulama her türlü veriyi HTML, BMP, GIF ve PNG dosyalarının içine saklayabilmektedir. Windows XP üzerinde çalışmaktadır. AES, BlowFish, veya 3DES şifreleme algoritmalarından biriyle gizlenecek veriyi şifreleme imkanı da sunmaktadır. Ticari olamayan amaçlar için ücretsiz olarak kullanılabilmektedir [SecureEngine].

#### 3.5.7.6. Hermetic Stego

Bu yazılım BMP dosya setinin içerisine her türlü veriyi saklayabilmektedir. Dolayısıyla yeterli miktarda BMP dosyası sağlanması durumunda saklanacak verinin boyutundan bağımsız çalışmaktadır. Hem şifreleme hem de saklama işleminde verinin rast gele dağıtımı için gizli bir anahtar kullanmaktadır [Hermetic Systems, 2006].

#### 3.5.7.7. GifShuffle

Windows üzerindeki komut penceresinden çalıştırılan bu uygulama animasyon ve şeffaflık içeren türler de dahil olmak üzere tüm GIF resimlerinin içerisine veri

saklayabilmektedir. Veri saklamadan önce aynı zamanda sıkıştırıp şifrelemektedir. Renk paletinde farklı permutasyonlar uygulama yöntemiyle çalışmaktadır [Gifshuffle].

#### 3.5.7.8. Revelation

Bu uygulama en az hata yöntemini kullanarak 24 bitlik BMP dosyalarının en az öneme sahip bitlerine veri saklamaktadır. Java ile yazılmıştır [Revelation].

#### 3.5.7.9. Deogol

Perl dilinde yazılmış olan bu uygulama komut penceresinden çalıştırılmaktadır. HTML dosyalarının içerisine taşıyıcı dosyanın boyutunu arttırmaksızın veri saklamaktadır [Deogol].

#### 3.5.7.10. InfoStego

Bu uygulama Windows üzerinde çalışmaktadır ve BMP dosyaları içerisine veri saklamaktadır. Veriyi saklamadan önce, saklama kapasitesini arttırmak için sıkıştırma, daha güvenli saklamak için de şifreleme imkânı sunmaktadır [Antiy Labs].

# 4. LSB YÖNTEMİNİ KULLANAN ALGORİTMALAR VE PROGRAMLAR

Son bite ekleme işlemi resmin başından ya da sonundan olmak üzere sıralı bir şekilde olabileceği gibi, bir rasgele fonksiyon üreteci (random function generator) kullanılarak belirlenen bir piksel üzerinde değişiklik yapılması şeklinde gerçekleştirilebilmektedir.

Bazı steganografik sistemler bazı gizli anahtarlar da kullanabilmektedir. Bu anahtarlar ikiye ayrılırlar:

1. *Steganografik anahtarlar;* mesajı resmin içine gizleme ve tekrar elde etme işlemini kontrol etme için kullanılırlar.
2. *Kriptografik anahtarlar;* Mesajın resmin içine gizlenmeden önce şifrelenmesi ve daha sonra deşifrelenmesinde kullanılırlar [Westfeld ve Pfitzmann, 1999].

Son bite ekleme yönteminin etkin bir biçimde kullanılabilmesi için görüntü dosyalarının özellikleri de son derece önemlidir. Bir görüntü dosyası Kayıpsız (lossless) ve Kayıplı (lossy) olmak üzere iki çeşide ayrılır.

Görüntünün kayıplı ya da kayıpsız olmasına göre LSB yöntemini kullanan çeşitli steganografik algoritmalar ve programlar geliştirilmiştir. Bunlardan bazıları aşağıda verilmişir.

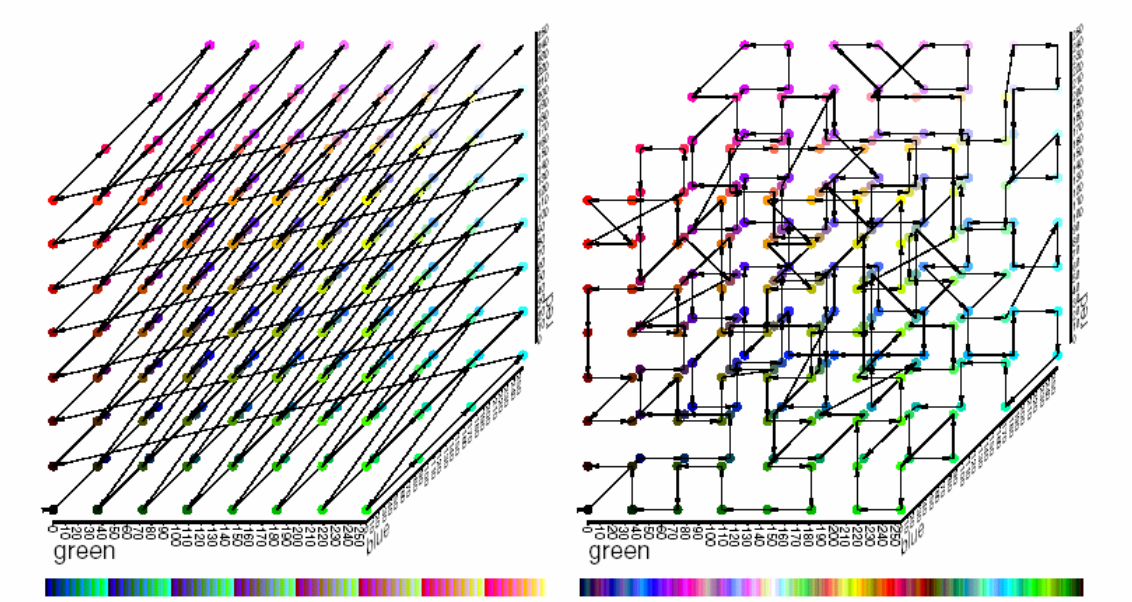
- EzStego
- S-Tools
- Hide and Seek
- J-Steg
- F4 ve F5
- Ayrık Logaritma Kullanan LSB Yöntemi

## 4.1. EzStego Algoritması

EzStego [EzStego] Romana Machado tarafından yazılmış olan ve GIF dosyalarına görüntü saklamak için kullanılan bir algoritma ve programdır [Westfeld ve Pfitzmann, 1999].

Öncelikle GIF dosyasının bir renk paleti matrisi oluşturulur. EzStego herhangi bir boyut bilgisi almadan mesajı piksellerin içine yerleştirir. Algoritma renk paleti matrisinin sıralı bir kopyasını yaratır. Bu sıralama işlemi öyle bir şekilde yapılmaktadır ki komşu renklerin birbiriyle farkları zorlukla söylenebilmektedir.

Üç boyutlu alanda bir nokta olarak her rengi RGB (Red, Green, Blue) renk kübü ile yorumlanabilir.




(a) (b)

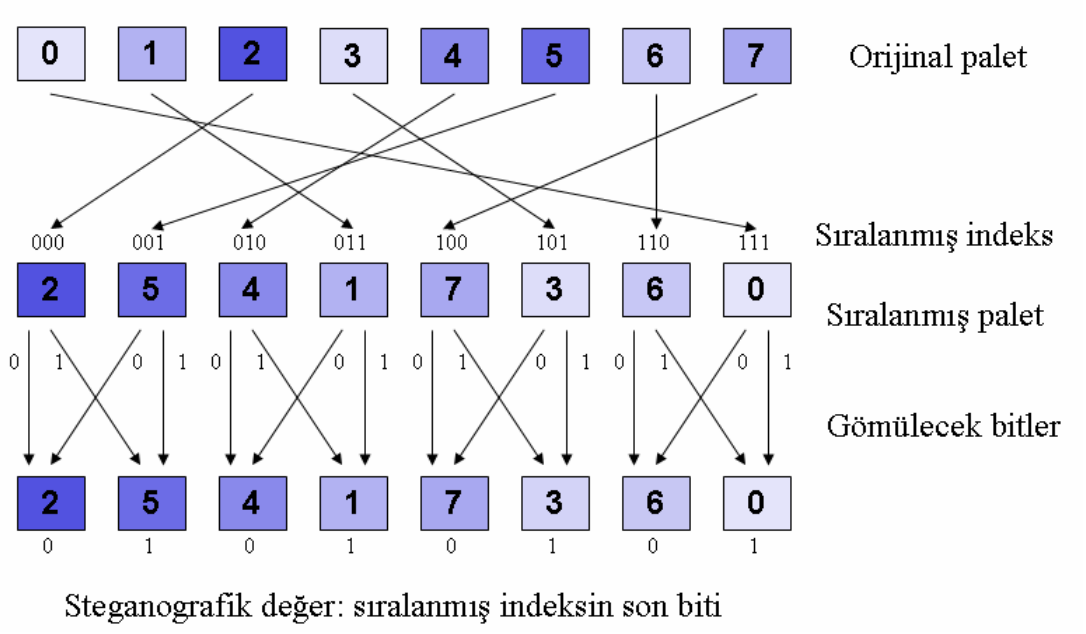
**Şekil 4.1. (a)** Palette renklerin sıralanışı **(b)** EzStego kullanılarak sıralanmış palet

Bu renk paletini sıralamak için şöyle bir yol izlenmektedir:

Listeden iki renk $\{c_0, c_1\}$ seçilerek başlanır. Buna $C$ kümesi denir. Aşağıdaki adımlar tüm renkler yerleşene kadar devam eder;

i. $C$ kümesinden en uzaktaki renk bulunur ve $d$ olarak isimlendirilir.
ii. Bu rengin yerleşeceği en iyi yer bulunur. Bu yer $\delta(C, c_i) + \delta(C, c_{i+1})$ değerini minimize edecek yerdir.
iii. $d$ yerleştirilir ve yeni $C$ kümesi olarak adlandırılır.

Gömme fonksiyonu resmin sol üst köşesinden başlayarak sağ alt köşeye kadar sırayı hiç bozmadan sıralı bir şekilde son bitleri değiştirir. Gömme işlemi sonucunda her pikselde bir steganografik değer bulunur. Her pikselin steganografik değeri, sıralanmış palette olması gereken son bitin indeksidir.




**Şekil 4.2.** Gömme fonksiyonu

## 4.2. S-Tools

S-Tools Andy Brown tarafından geliştirilen ve GIF ve BMP dosyaları üzerinde çalışan bir programdır [Brown, 1996].

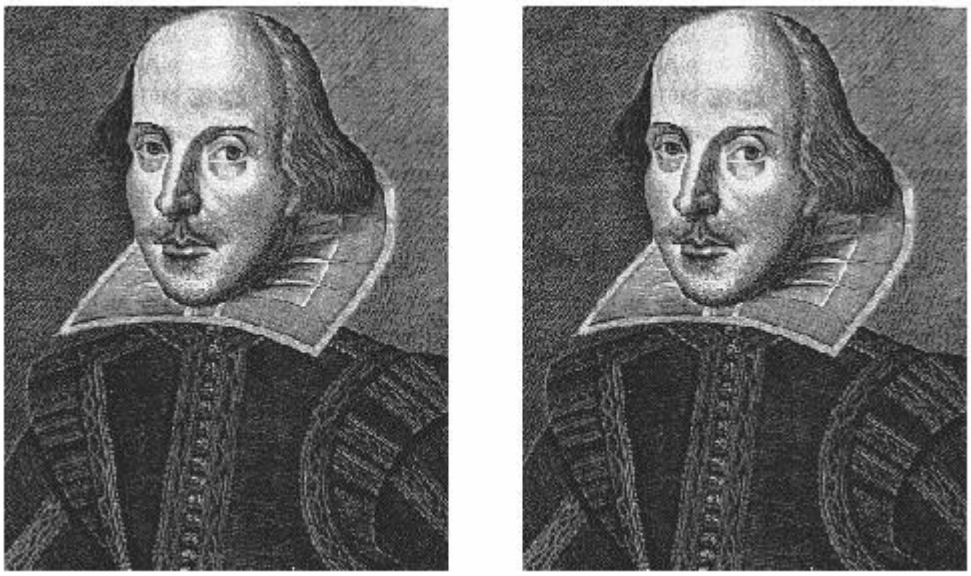
S-Tools programı gri seviye ve 24 bit renkli resimler üzerinde çalışmaktadır. Ayrıca bu program WAV formatındaki ses dosyalarının içinde de bilgi gizlemek için kullanılmaktadır.

Mesajın bitleri bir rasgele numara üreteci yardımıyla resmin içine saklanmaktadır. Kullanılan bu rasgele numara üreteci MD5 [Rivest, 1992] algoritmasına dayanmaktadır.

Saklanacak metnin öncelikle IDEA [Lai, 1992], DES, 3DES [FIPS 46-3, 1999] gibi yöntemlerle şifrelenerek LSB yöntemine göre resmin içine yerleştirilmesini öngörür.

S-Tools mesajı saklamadan önce mesajın önüne ekstra bilgi eklemesi yapar. Zamana bağlı rastlantısal 32 bitlik bir mesaj özeti ilk önce eklenir. Bu adımın anlamı CBC (Cipher Block Chaining) ya da PCBC (Propagating Cipher Block Chaining) modunda şifrelenmiş iki aynı dosyanın aynı şifreli metne şiflenemeyecek olmasıdır. Bu 32 bit uzunluğundaki özet gizli dosyaya dâhil edilir. Bu S-Tools'un gizli dosyayı tekrar elde etmesi için gereklidir. Şifreleme bu değeri gizleyecektir.

Şekil 4.3'te orijinal resim ve S-Tools programıyla içine bilgi gizlenmiş resim verilmiştir [Johnson, 1996].



**Şekil 4.3**. Orijinal ve içine S-Tools ile bilgi gizlenmiş resim

## 4.3. Hide and Seek

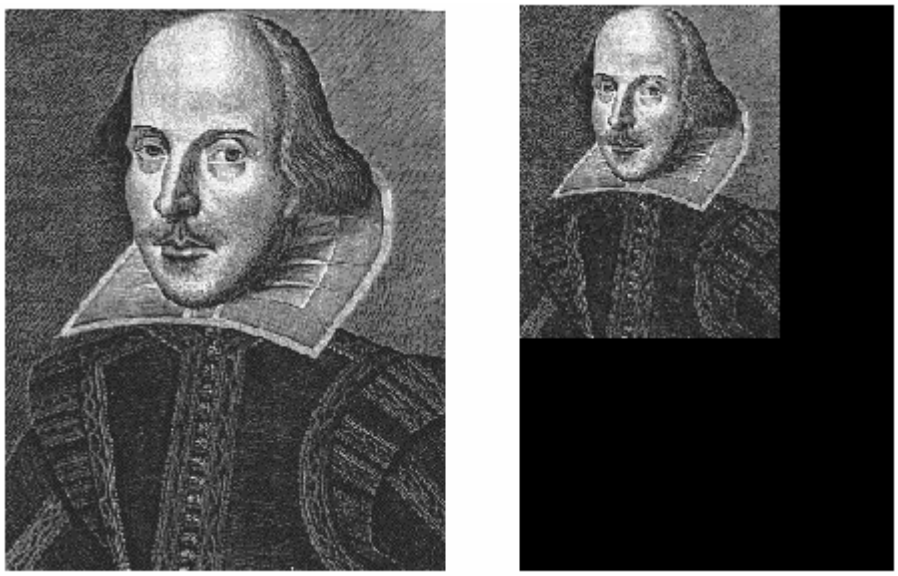
Hide and Seek, Colin Maroney tarafından geliştirilen ve GIF formatındaki dosyalar üzerinde DOS ortamında çalışan bir steganografi programıdır [Maroney].

Bu yöntemde rastgele LSB yöntemine göre bilgi saklamaktadır. Görüntü dosyasının boyutu 320x480 piksel olarak belirlenmiştir. Bu boyuttan daha ufak olan resimler için, içine bilgi gizlendiğinde yeni oluşan dosyanın bir kısmı siyah olarak gözükmektedir (320x480 piksel boyutuna tamamlamak için). Eğer görüntü dosyasının boyutu 320x480 pikselden büyükse, stego-nesnesi kesilir ve uygun hale getirilir.

Görüntü dosyasının 1024x768 pikselden büyük olması durumunda ise program çalışmamaktadır.

Görüntü dosyasının maksimum büyüklüğü 320x480 piksel olabileceği için, içine gizlenebilecek bilgi miktarı da gri seviye resimlerde en fazla 19.000 byte'dir.

Aşağıda 222x282 piksel boyutundaki 256 bit gri seviye Shakespeare resmi ve M1 mesajının Hide and Seek programıyla bu resmin içine gizlenmesiyle elde edilen 340x480 piksel boyutundaki stego-resim verilmiştir [Johnson, 1996].



**Şekil 4.4.** Orijinal Shakespeare resmi ve içine M1 mesajının Hide and Seek programıyla bilgi gizlenmesiyle elde edilen resim

Şekil 4.4'ten de görülebileceği gibi orijinal resim belirlenen değerden küçük olduğu için 340x480 piksele tamamlamak için siyah bir alan eklenmiştir.

## 4.4. J-Steg

J-Steg algoritması Derek Upham tarafından geliştirilmiş bir algoritmadır ve bu algoritmayı kullanan birçok program geliştirilmiştir [Upham, 2003]. J-Steg JPEG dosyalara son bite ekleme yöntemine göre bilgi gizleyen bir algoritmadır.

JPEG dosyaları, RGB renk gösterimini YUV (YCbCr) şekline dönüştürür ve buna göre her 8x8 piksellik blok için bir DCT (Discrete Cosine Transform) değeri hesaplar. Daha sonra bir nicelendirme matrisi yardımıyla nicelendirme işlemi yapılarak nicelendirilmiş DCT katsayı matrisi hesaplanır.

Daha sonra yok edilecek frekans katsayılarını dikkate alarak her bir blokta 1 byte saklarlar. Burada önemli olan, düşük frekans değişimlerinin resmi bozacağının kesin olmasıdır. Öte yandan, yüksek frekans katsayıları da muhtemelen yok edileceklerdir.

Dolayısıyla resmi değiştirmeyecek kadar yüksek, yok edilmeyecek kadar düşük frekans katsayılarının bulunması gerekir. Ayrık kosinüs dönüşümü kullanarak gizli iletişimi sağlayan yöntemlerde ortak sorun katsayıların belirlenmesidir. Ancak, katsayılar resimden resme fark göstermektedir. Resimde bozulma olup olmayacağıda maalesef ancak işlem sonrası anlaşılabilmektedir. Ayrıca, her 64 adetlik bloklara 1 byte gömülebilmesi, saklanabilecek veri miktarını önemli oranda düşürmektedir.

JPEG dosyalar üzerinde işlem yaparken şu aşamalar uygulanır [Wallace, 1991]. Jsteg programı da aşağıdaki işlemleri gerçekleştirmektedir.

1. Resim RGB renk formatını kullanıyor ise YUV (YCbCr) dönüşümü uygulanır
2. Resim 8x8 piksellik bloklara bölünür
3. Her blok için DCT katsayı matrisi hesaplanır
4. Quantization (Nicelendirme) işlemi yapılır.

**RGB→YUV(YCbCr) Dönüşümü:** Renkli resimler için JPEG dosya özelliğinden dolayı bu işlemin yapılması gerekmektedir. Resim Gri-Seviye'li bir resim ise bu işlem uygulanmaz.

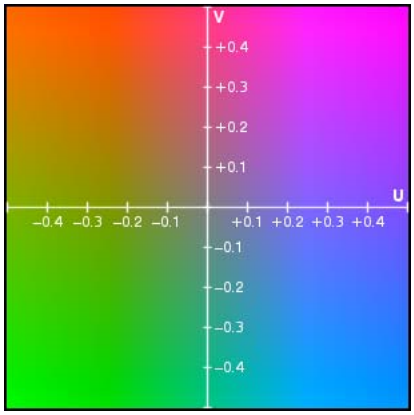
RGB→YUV dönüşümü bir lineer dönüşümdür ve bu dönüşüm için Eric Hamilton tarafından tarif edilmiş olan denklem ve ters dönüşüm denklemi aşağıda verilmiştir [Hamilton, 1992].

Dönüşüm şu formüller yardımıyla yapılır. Burada Y: İntensity (Parlaklık), Cb: Blue/Yellow (Mavi/Sarı), Cr: Red/Green (Kırmızı/Yeşil) göstermektedir.

$$Y = + 0.299R + 0.587G + 0.114B$$

$$Cb = + 0.492(B - Y) = - 0.147R - 0.289G + 0.436B$$

$$Cr = + 0.877(R - Y) = + 0.615R - 0.515G - 0.100B$$




**Şekil 4.5.** Y değeri için U-V grafiği

Şekil 4.6. (a)'da verilen orijinal resim üzerinde RGB-YCbCr dönüşümü uygulandığında elde edilen sonuçlar Şekil 4.6.(b-c-d)'de verilmiştir.



**Şekil 4.6. (a)** Orijinal RGB resim **(b)** Y(Parlaklık) **(c)** Cb (mavi/sarı) bileşeni **(d)** Cr (kırmızı/yeşil) bileşeni

**DCT Katsayı Matrisinin Hesaplanması:** DCT; piksel değerlerinin -128 ile 127 arasında çalışması esasına dayandığı için öncelikle orijinal görüntü piksellerinden 128 değeri çıkartılır ve böylelikle M matrisi elde edilir.

8×8 piksellik görüntü blokları için, her blok diğerinden bağımsız olarak ayrık kosinüs dönüşümüne (DCT) tabi tutulur. Sonuçta ortaya yine 64 katsayı çıkacaktır. Bunlardan ilkine DC (ilk piksel - P(0,0)) ve diğerlerine de AC adı verilir. DCT dönüşümünün formülü aşağıda verilmiştir [Cabeen ve Gent, 1998]:

$$D_{ij} = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \tag{4.1}$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if} \quad u = 0 \\ 1 & \text{if} \quad u > 0 \end{cases} \tag{4.2}$$

**Nicelendirme İşlemi:** DCT matrisi elde edildikten sonra bir nicelendirme işlemi yapılır. Bunun için bir nicelendirme matrisi gereklidir. Nicelendirme matrisi istenilen kaliteye göre değişmektedir, 50 kalite faktörü için kullanılan matris aşağıda verilmiştir.

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Nicelendirme işlemi aşağıdaki formül kullanılarak yapılır.

$$C_{i,j} = round\left(\frac{D_{i,j}}{Q_{i,j}}\right) \tag{4.3}$$

Nicelendirilmiş katsayı matrisi hesaplandıktan sonra Jsteg programı ilk 8x8 piksellik bloğu 3 sahaya ayırır ve LSB yöntemini kullanarak bilgiyi gizler.




A sahası 5 bit uzunluğundadır ve kendinden sonraki kaç bitin mesaj uzunluğunu belirlemek için alınacağını belirler.

A sahasından hesaplanan değer B sahasının uzunluğudur ve gizlenen mesajın uzunluğunu vermektedir.

Daha sonra gelen 8 bit ise C sahasıdır ve mesajın ilk karakterini içermektedir.

## 4.5. F5 Algoritması

F5 algoritması Andreas Pfitzmann ve Andreas Westfeld tarafından 2001 yılında ortaya çıkartılmıştır [Westfeld, 2001]. Yöntemin amacı, JPEG dosyalardaki saklama kapasitesini güvenlikten feragat etmeden arttırmaktır. Bu yöntem, yine Westfeld tarafından geliştirilmiş olan $\chi^2$ testi öncülüğünde mesaj bitlerinin nicelendirilmiş DCT katsayılarının LSB'leri ile değiştirmek yerine katsayıların mutlak değerinin 1 azaltılmasına dayanır. Yazarlar bu tür gömme işlemlerinin $\chi^2$ istatistiksel saldırısı kullanılarak sezilemeyeceğini öne sürmektedirler.

F5 algoritması mesaj bitlerini rasgele seçilen DCT katsayılarına gizlerler. Belirli bir uzunluktaki mesajı gömmek için gerekli değişimlerin sayısını minimize eden bir matrissel gömme işlemi uygulamaktadır.

Gömme işlemi esnasında; mesaj uzunluğu, 0 ve DC (DCT matrisinin 0x0 indeksli elemanı) olmayan katsayıların sayısı, örtü verisindeki (cover image) değişikliklerin sayısını minimize eden en iyi matris gömme işlemini elde etmek için kullanılır. Matris gömme 3 parametreden $(c,n,k)$ oluşur. Burada $c$, her gruptaki $n$ katsayılarının değişim sayısı, $k$ ise gömülecek bitlerin sayısıdır. Örneğin basit bir

matris gömme $(1,2^k-1,k)$ işleminde bir hash (anahtarlama) fonksiyonu $2^k-1$ katsayısına uygulandığında $k$ bitlik çıkış verecektir [Westfeld, 2001].

Gömme işlemi kullanıcı şifresinden bir PRNG (Pseudo Random Number Generator – Sahte Rasgele Numara Üreteci) için bir taslak elde etme işlemi ve örtü verisinin DCT katsayıları yolu ile bir rasgele yürüyüş elde etmesiyle başlamaktadır. PRNG aynı zamanda $k$ değerinin bir akış şifre kullanılarak şifrelenmesi ve bu değerin mesaj dizisinin başında mesaj uzunluğu ile beraber kurallı bir şekilde gömülmesi için kullanılır. Mesaj gövdesi matris gömme işlemi kullanılarak gömülür. Bu gömülme işlemi sırasında $k$ mesaj biti $2^k-1$ katsayılı bir gruba eklenir, bu işlem ise her grubun en yüksek mutlak değere sahip olan katsayısı bir azaltılarak gerçekleştirilir.

Gömme işlemi aşağıdaki 6 adımı içerir.

**Adım 1:** Giriş görüntüsünün RGB gösterimini elde et.

**Adım 2:** Nicelendirme tablosunu ve bunu takiben $Q$ kalite faktörünü hesapla. Nicelendirilmiş DCT katsayılarını saklarken görüntüyü de sıkıştır.

**Adım 3:** Matris gömme işlemi olmaksızın tahmini kapasiteyi $C = h_{DCT} - h_{DCT}/64 - h(0) - h(1) + 0{,}49h(1)$ formülünü kullanarak hesapla. Burada $h_{DCT}$ tüm DCT katsayılarının sayısı, $h(1)$ DCT matrisinin kesin değeri 1 olan AC (DCT katsayı matrisinin 0x0 indeksli elemanı hariç diğer elemanları) katsayılarının sayısını belirtmektedir. $h_{DCT}/64$ ise DC katsayılarının sayısını, $-h(1)+0{,}49h(1) = -0{,}51h(1)$ ise fireden dolayı tahmin edilen kaybı göstermektedir.

**Adım 4:** Kullanıcı tanımlı şifre, mesaj bitlerini gömmek için bir rasgele yol elde eden PRNG için bir taslak üretmek amacıyla kullanılır. PRNG aynı zamanda bir sahte rast gele bit dizisi elde etmek için kullanılır. Bu sahte rast gele bit dizisi rastlantısal bit dizisi elde etmek için mesaj ile XOR'lanır. Gömme işlemi boyunca DC katsayıları ve 0'a eşit olan katsayılar göz ardı edilir.

**Adım 5:** Rastgele yol boyunca $2^k - 1$ sayılık bir gruba, $k$ bitlik segmentlere bölünen mesaj gömülür. Eğer bu grubun hash'i mesaj bitlerine uyum sağlamaz ise gruptaki katsayılardan birinin mutlak değeri bir eşleşme elde etmek için bir azaltılır. Eğer katsayı 0 olursa bu olaya fire (shrinkage) adı verilir ve aynı $k$ mesaj biti DCT katsayılarının bir diğer grubuna tekrar gömülür. $LSB(d) = d \bmod 2,\ d > 0\ icin$, ve $LSB(d) = 1 - d \bmod 2, d < 0\ icin$

**Adım 6:** Eğer mesaj boyutu tahmini kapasiteye uygunsa, gömme işlemi ilerler, aksi halde maksimum mümkün uzunluğu gösterecek şekilde bir hata mesajı verilir. Tahmin edilen firenin daha fazla olması nedeniyle de kapasite yüzünden bazı ender rastlanan sonuçlar ortaya çıkabilir. Böyle bir durumda, gömülebilecek en uygun miktarı belirten bir mesaj gösterilir.

F5 algoritması, PoV'ler üzerinde değişiklik yapmadığından dolayı $\chi^2$ testi ile keşfedilememektedir [Fridrich vd., 2002].

## 4.6. Ayrık Logaritma Kullanan Rasgele LSB Ekleme Yöntemi

Sıralı LSB yönteminde verinin resmin satırlarına ya da sütunlarına sıra ile yerleştirilmesinden dolayı gizli mesajın elde edilmesi işlemi oldukça kolaydır. Bu yüzden verileri rasgele bir şekilde resmin içine saklamak daha güvenlidir. Ayrık logaritma fonksiyonu kullanarak veri gizleme M. M. Amin, M. Salleh, S. Ibrahim, ve M. R. Katmin tarafından geliştirilmiş bir yöntemdir [Amin vd., 2003].

$y_i = a^i \bmod p$ şeklinde tanımlanan ayrık logaritma fonksiyonu resim içine rasgele şekilde veri gizlemeyi sağlar. Burada $y_i$, mesajın $i$. bitinin resmin içinde

saklanacağı pozisyonu; $i$ gizlenecek mesajın bit indeksini göstermektedir. Buradaki $p$ büyük bir asal sayı ve $a$ ise $p$'den üretilen asal bir köktür. $a$ değeri üsler şeklinde yazıldığında 1'den $p-1$'e kadar tüm tamsayıları verecek şekilde seçilmelidir. Yani $p$ ile $a$ kendi aralarında asal olmalıdır. $p$ değerinin asal olmasının nedeni aynı değerin tekrar üretilmemesidir.

$p$ değeri seçildikten sonra $a$ değerinin üretilmesi için sayı teorisinden yararlanılmaktadır. Aşağıda bunun için kullanılan tanım ve teoremler verilmektedir.

**Teorem 1:** Eğer $p$ asal sayı ise $Z_p$ bir alandır.

**Teorem 2:** Eğer $p$ asal sayı ise $Z_p^*$ çevrimsel bir gruptur.

**Tanım 1:** $Z_p$ alanının 0 olmayan bir elemanı olan $\alpha$'nın derecesi $\alpha^k = 1$ olmak üzere en küçük $k$ değeridir.

**Tanım 2:** $\bmod p$'ye göre $(p-1)$ derecesine sahip bir $\alpha$ elemanına asal eleman denir.

**Tanım 3:** $p$ asal ve $\alpha$, $\bmod p$'ye göre asal eleman olsun. Herhangi bir $\beta \in Z_p^*, \beta = \alpha^i \ (0 < i < p-2)$ olmak üzere yazılabilir. $\beta = \alpha^i$'nin derecesi $\dfrac{p-1}{OBEB(p-1,i)}$'dir. Böylece eğer $OBEB(p-1,i) = 1$ ise $\beta$ asal bir elemandır.

**Teorem 3:** $p$ asal ve $\alpha \in Z_p^*$ olsun. O zaman $\alpha \bmod p$'ye göre eğer $\alpha^{\frac{p-1}{q}} \neq 1 \pmod p$ ise $(p-1)$'i bölen tüm $q$ değerleri için $\alpha$, $\bmod p$'ye göre asaldır.

Gizlenecek metnin uzunluğu $m$, içine veri gizlenecek resmin büyüklüğü $l$ ise $p$ değeri, $m < p < l$ şartını sağlamalıdır.

Veri gizleme işleminin algoritması şu şekildedir.

**Adım 1:** $m < p < l$ şartını sağlayan en büyük asal sayıyı $p$ olarak seç.

**Adım 2:** $p$ 'nin asal elemanları sayısını $(\phi)$ bul.

**Adım 3:** Asal elemanları üretmek için en küçük böleninden başlayarak üsler şeklinde yazıldığında 1'den $p-1$'e kadar tüm tamsayıları veren böleni bul.

**Adım 4:** $OBEB(i, p-1) = 1$ şartını sağlayan $i$ değerlerini bul.

**Adım 5:** mod $p$ 'ye göre $bölen^i$ değerlerini hesapla ve büyük olanlardan birini $a$ olarak seç.

**Adım 6:** $y_i = a^i \bmod p$ denklemine göre mesajın bitlerinin hangi piksellere yerleşeceğini bul.

Örnek olarak $p$ değeri 17 olarak seçilsin. Öncelikle $p$ değerinin kaç tane asal elemanı $(\phi)$ olduğunu hesaplanır. Bunun için öncelikle $p-1$ değeri çarpanlarına ayrılır ve üslü şekilde yazılır.

$p-1$'in çarpanları $16 = 2 \times 2 \times 2 \times 2 = 2^4$

$\phi = 2^4 - 2^3 = 16 - 8 = 8$

$p$ değerinin 8 adet asal elemanı olduğu bulunur. $p$ değerinin asal elemanlarını bulmak için aşağıda gösterildiği gibi, en küçük böleninden başlayarak üsler şeklinde yazıldığında 1'den $p-1$'e kadar tüm tamsayıları veren en küçük değer bulunur.

$2^0 = 1 \mod 17$ $3^0 = 1 \mod 17$
$2^1 = 2 \mod 17$ $3^1 = 3 \mod 17$
$2^2 = 4 \mod 17$ $3^2 = 9 \mod 17$
$2^3 = 8 \mod 17$ $3^3 = 10 \mod 17$
$2^4 = 16 \mod 17$ $3^4 = 13 \mod 17$
$2^5 = 15 \mod 17$ $3^5 = 5 \mod 17$
$2^6 = 13 \mod 17$ $3^6 = 15 \mod 17$
$2^7 = 9 \mod 17$ $3^7 = 11 \mod 17$
$2^8 = 1 \mod 17$ $3^8 = 16 \mod 17$
$2^9 = 2 \mod 17$ $3^9 = 14 \mod 17$
$2^{10} = 4 \mod 17$ $3^{10} = 8 \mod 17$
$2^{11} = 8 \mod 17$ $3^{11} = 7 \mod 17$
$2^{12} = 16 \mod 17$ $3^{12} = 4 \mod 17$
$2^{13} = 15 \mod 17$ $3^{13} = 12 \mod 17$
$2^{14} = 13 \mod 17$ $3^{14} = 2 \mod 17$
$2^{15} = 9 \mod 17$ $3^{15} = 6 \mod 17$
$2^{16} = 1 \mod 17$ $3^{16} = 1 \mod 17$

Bölen olarak 2 seçildiğinde 1'den $p-1$'e kadar tüm tamsayıları vermemekte ve aynı zamanda tekrarlar olmaktadır. 3 seçildiğinde ise istenen şart sağlanmaktadır. $\mod 17$'ye göre $3^i$ değerleri $p$ ile asaldır.

Bir sonraki adım $OBEB(i, p-1) = 1$ şartını sağlayan sayıları bulmaktır. Bu değerler bulunduktan sonra $3^i$'de karşılık gelen değerler hesaplanır ve bu bulunan değerlerden biri (tercihen büyük olanı) $a$ değeri olarak seçilir.

$OBEB(i, p-1) = 1$ şartını sağlayan sayılar 1, 3, 5, 7, 9, 11, 13 ve 15'tir. mod 17'ye göre $3^i$'de karşılık gelen değerleri aşağıdaki gibi bulunur.

$3^1, 3^3, 3^5, 3^7, 3^9, 3^{11}, 3^{13}, 3^{15}$

$a \rightarrow 3, 10, 5, 11, 14, 7, 12, 6$

$a$ değerleri arasından büyük olanların seçilmesi daha uygundur. Burada 14 seçilebilir.

$p$ ve $a$ değerleri seçildikten sonra $y_i = a^i \bmod p$ denklemine göre gizli mesajın bitlerinin hangi piksellere yerleşeceği sırasıyla bulunur ve LSB yöntemine göre bu piksellere yerleşir.

**Örnek:**

Şekil 4.7 (a)'da verilen, boyutu 182x175 piksel olan örnek resmin içine **SELAM** mesajı gizlenmek istensin [Şahin vd., Şubat 2006]. Resmin büyüklüğü $182x175 = 31850$ 'dir. Metnin uzunluğu ise 40 bittir.

$m < p < l$ şartını sağlayan en büyük asal sayı $p$ değeri olarak seçilir.

$$p = 31849$$

$p$ değerinin kaç tane asal elemanı ($\phi$) olduğu hesaplanır. Bunun için $p-1$ değerini çarpanlarına ayrılır ve üslü şekilde yazılır.

$$p-1 = 31848 = 2^3 \times 3^1 \times 1327^1$$

$$\phi = (2^3 - 2^2)(3^1 - 3^0)(1327^1 - 1327^0) = 10608$$

adet asal elemanı olduğu bulunur.

$p$ değerinin asal elemanlarını bulmak için en küçük böleninden başlayarak üsler şeklinde yazıldığında 1'den $p-1$'e kadar tüm tamsayıları veren böleni bulunur. Burada bu değer 11 olarak hesaplanmaktadır. Sonraki adımda $OBEB(i, p-1) = 1$ şartını sağlayan sayılar bulunur ve bunların $11^i \pmod p$ 'de karşılık gelen değerlerinden büyük bir $a$ değeri seçilir. Burada $11^7 = 27432 \pmod{31849}$ 'dan $a = 27432$ olarak seçilmiştir. $p$ ve $a$ değerleri belirlendikten sonra $y_i = 27432^i \pmod{31849}$ denkleminden mesajın bitlerinin sırasıyla hangi piksellere yerleşeceği bulunur. Mesajın ilk harfinin (S-01010011) hangi piksellere yerleştiği ve piksellerde meydana gelen renk değişiklikleri aşağıdaki tabloda gösterilmektedir.

**Tablo 4.1.** Mesajın ilk harfinin resim içinde yerleşim yerleri

| Mesajın bit indeksi | Bit değeri | Yerleştiği piksel | Pikselin orijinal rengi | Pikselin yeni rengi |
|---|---|---|---|---|
| 1 | 0 | 27432 | 120 | 120 |
| 2 | 1 | 18301 | 97 | 97 |
| 3 | 0 | 27409 | 124 | 124 |
| 4 | 1 | 2517 | 150 | 151 |
| 5 | 0 | 29651 | 141 | 140 |
| 6 | 0 | 9963 | 89 | 88 |
| 7 | 1 | 8747 | 40 | 41 |
| 8 | 1 | 29187 | 200 | 201 |

Bu şekilde mesajın tüm bitleri resmin içine rasgele bir şekilde dağılarak gizlenmektedir.



(a) (b)

**Şekil 4.7. (a)** Orijinal resim (lenna.bmp) **(b)** İçine **SELAM** mesajı gizlenmiş resim

# 5. STEGANOGRAFİK SİSTEMİ DEĞERLENDİRME KRİTERLERİ

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

- Taşıyıcıdaki değişim
- Kapasite
- Dayanıklılık'tır.

## 5.1. Taşıyıcıdaki Değişim

Bir steganografik algoritma değerlendirilirken taşıyıcıda (cover object) ne kadar değişim olduğu da çok önemlidir. Taşıyıcıdaki değişimi yada resimdeki bozulma oranının belirlenmesi için çeşitli ölçme yöntemleri vardır. Bunlar arasında en bilinenleri; MSE, RMSE, PSNR'dır.

MSE (mean squared error) hataların kareleri toplamının ortalamasıdır. MSE genellikle $\sigma^2$ olarak gösterilir. RMSE (root mean squared error) ise MSE'nin karekökü'dür [Sayood, 1996].

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2 \tag{5.1}$$

Bazen MSE yerine, hatanın büyüklüğünün orijinal piksel değerinin en büyüğü (peak-tepe) ile olan ilişkisi ile ilgilenilir. Bu gibi durumlarda PSNR (peak signal-to-noise ratio) yöntemi kullanılmaktadır [Sayood, 1996].

$$PSNR(dB) = 10\log_{10}\frac{x_{peak}^2}{\sigma_d^2} \tag{5.2}$$

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

## 5.2. Kapasite

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

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

## 5.3. Dayanıklılık

Bir steganografik sistemin dayanıklılığını ölçmek için steganaliz yöntemleri kullanılmaktadır.

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

Genelde saldırı yapan kişinin (steganalist) kullanılan steganografik sistemi bildiği varsayılır (*Kerchoffs'un prensibi*). Eğer steganalist kullanılan sistemi bilmiyorsa, bu onun işini zorlaştıracaktır [Rijmen, 1997]. Steganalistin bir steganografik sisteme saldırabilmesi için sahip olması gereken veriler vardır. Bu sahip olduğu verilere göre saldırı modellerinden birini seçebilir.

Bu saldırı modellerinden en yaygın olanları şunlardır [Biryukov, 1999] [Rijmen, 1997] [Stinson, 2002] :

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

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

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

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

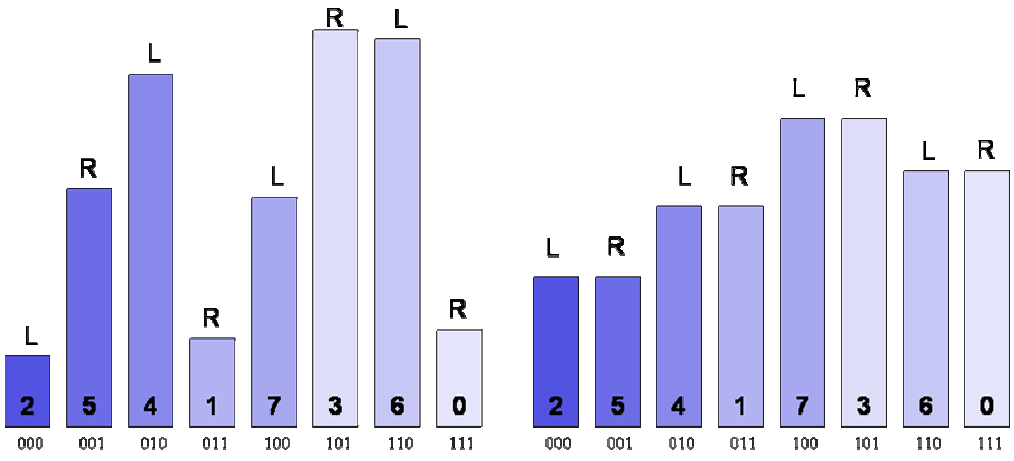
şeklinde sınıflandırılabilir.

Resmin içinde veri olduğu anlaşılırsa, bu veriyi elde etmek amacıyla çekme (extraction) saldırısı yapılır [Phan ve Ling, 2003].

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

### 5.3.1 $\chi^2$ Testi

LSB içinde PoVs (pairs of values) değerlerinin istatistiksel analizi temelli olan $\chi^2$ istatistik testi, Westfeld tarafından sunulan bir steganaliz metottur [Westfeld ve Pfitzmann, 1999]. PoVs'ler piksel değerlerinden meydana getirilebilir, ölçülmüş DCT katsayıları veya palet indisleri sadece LSB'de değişmektedir. İçine veri gizlenmemiş görüntüler için PoVs'lerin frekansları düz bir şekilde dağılmamaktadır, fakat LSB gizleme steganografi söz konusu olunca her PoVs' in frekansları eşit olmaktadır. Şekil 5.1 mesajın gizlenmesinden önce ve sonraki renk histogramını göstermektedir.




**Şekil 5.1.** Bir resmin içine gizli mesaj gömülmeden önceki ve sonraki renk dağılım histogramı.

Bundan sonra Westfeld şüpheli bir görüntüde ölçülen PoVs'lerin oluşunu, istatistiksel rasgele testi ile karşılaştırmıştır.

$\chi^2$ istatistik testin ayrıntıları aşağıda verilmiştir:

**Adım 1:** $k$ kategoriler ve gözlemlerden oluşan rasgele bir örnekleme olduğunu varsayılmakta. Her gözlem sadece ve sadece bir kategoriye düşmektedir. Şüpheli bilginin PoVs'lerinin tek değerlerine önem verilmektedir.

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

$$n_i^* = \frac{\left|\left\{renk \middle| (renk)'in\ siralanmis\ indeksi \in \{2i, 2i+1\}\right\}\right|}{2}$$

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

$$n_i = \left|\left\{renk \middle| (renk)'in\ siralanmis\ indeksi = 2i\right\}\right|$$

**Adım 4:** $\chi^2$ istatistiği ise k-1 bağımsızlık dereceleri ile şu şekilde hesaplanır:

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

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

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

$\chi^2$ istatistiksel analizi sıralı LSB gömme steganografide başarılı sonuç vermiştir.

Provos ise bu metodu test aralıkları ve değerleri yeniden değiştirerek genişletmiştir. Provos, test aralığı ve piksel değerlerini $P$ ve $(P+1)$ piksel çiftinden $P$ ve $(P-1)$ çifte kadar yeniden değiştirip, $\chi^2$ testini geliştirmiştir [Provos, 2001].

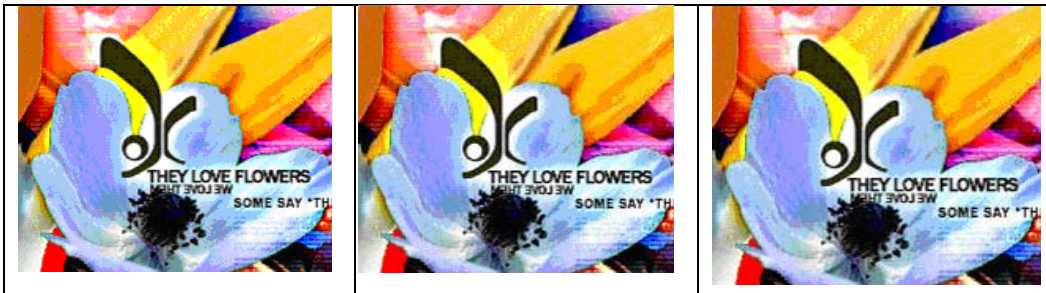
### 5.3.2. Histogram Analizi (PoVs'lerin Analizi)

PoVs (pairs of values)'ler, piksel değerlerinden, ölçülmüş DCT katsayılarından veya LSB'de değişen palet indislerinden oluşturulabilmektedir. Gizlemeden önce taşıyıcı görüntüde her çiftten iki değeri eşit olarak dağılmamaktadır. Mesaj gizlemesinden sonra, her çiftte değerlerin var oluşu eşit olmaya yönelik olacaktır. Bu analizde $\chi^2$ testi kullanılmaktadır. Her çiftte değerlerin var oluşu eşit olmasının istatistiksel anlamı test edilebilmektedir. Mesela bir palet görüntü için, palet içinde en fazla 256 renk($c_i$) vardır, bu demek ki en fazla 128 PoVs vardır.

Sıralı bir şekilde saklanmış bir mesaj için, mesajın saklama sırası ile aynen mesajı tarayabilmekte ve bütün piksel değerleri için $p$ değeri hesaplanmaktadır. İlk başta $p$'nin değeri 1'e yakın olacak ve sonra mesajın sonu geldiğinde hemen 0'a düşecektir. Bu en düşük sağ köşesine var olana kadar 0 olarak kalacaktır. Böylece bu test bir mesajın gizleme olasılığını belirtip gizli mesajın büyüklüğünü da belirtmektedir.

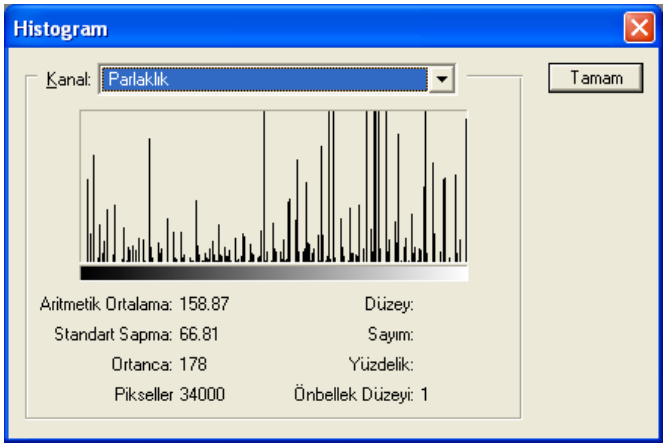
Eğer görüntüde mesaj taşıyıcı pikseller rasgele seçilirse bu test daha az etkili olmaktadır. Provos, bu tekniğin bir görüntünün daha küçük farklı alanlarına uygulanırsa, mesaj uzunluğu arttıkça $p$'nin değeri gitgide azaldığını fark etmiştir. Fakat Provos bununla ilgili daha ileri istatistiksel analizleri sunmamıştır.

Şekil 5.2'de verilen ve içine 1 KB, 5 KB saklanmış olan 200x170 piksel boyutlarındaki resim için histogram sonuçları sırasıyla Şekil 5.3, Şekil 5.4 ve Şekil 5.5 aşağıda verilmiştir.

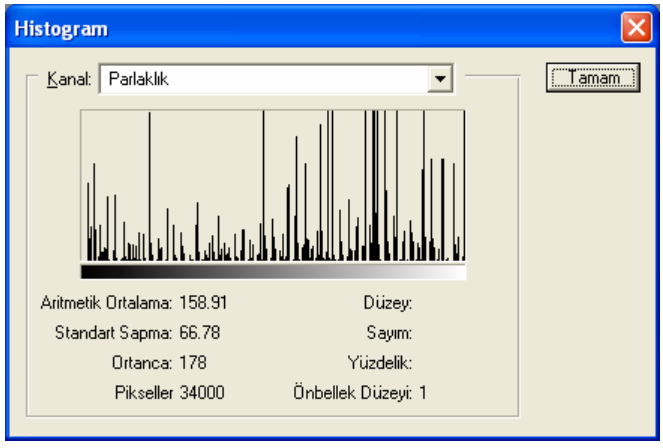



| (a) | (b) | (c) |
|---|---|---|

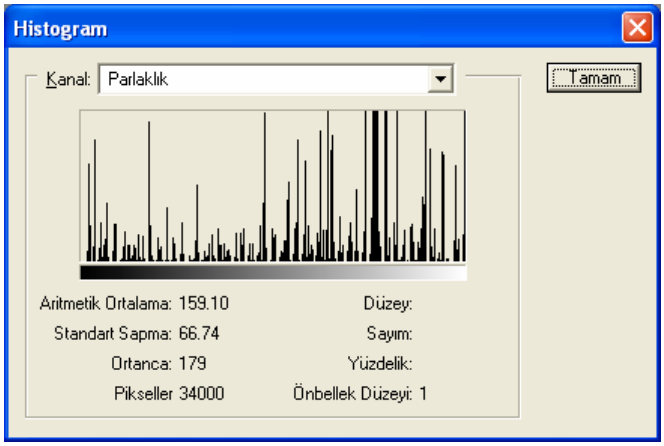
**Şekil 5.2. (a)** Orijinal resim (200x170 piksel) **(b)** 1 KB saklanmış resim **(c)** 5 KB saklanmış resim




**Şekil 5.3.** Orijinal resim için histogram




**Şekil 5.4.** 1 KB veri saklanmış resmin histogramı




**Şekil 5.5.** 5 KB veri saklanmış resmin histogramı

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

### 5.3.3. RS Steganaliz (İkili İstatistik Yöntemi)

Bu analiz, görüntülerde uzaysal korelâsyonlardan üretilen duyarlı ikili istatistiklerini kullanmaktadır. RS steganalizi 24 bit renkli ve 8 bit gri seviye görüntülerde kullanılmaktadır. RS steganalizinde, bir görüntünün piksellerinin 3 bağımsız gruba: Düzenli (Regular-R), Tekil (Singular-S) ve Kullanılmayan (Unused-U) olarak ayrılması esastır. Fridrich tarafından geliştirilmiştir [Fridrich ve Goljan, 2002]. Sayısal kamera veya tarayıcı ile alınan yüksek kaliteli görüntüler için, RS steganalizi güvenli bit-oranın her örnekleme için 0.005 bitten küçük olduğunu göstermektedir.

RS' yi açıklamak için birkaç notasyona ihtiyaç duyulmaktadır.

Test edilen görüntü $(R)$, $P$ kümesinden değer alan $MxN$ piksel'den oluşmaktadır. Örnek olarak, 8-bit gri seviyeli bir görüntüde, $P = \{0,....,255\}$ 'tir. Sonra $R$, $n$ komşu pikselden oluşan $G$ ayrı gruba bölünmektedir:

$$G = (x_1,....,x_n) \in R \qquad (5.3)$$

Ayrıcı fonksiyon aşağıdaki gibi belirlenmiştir:

$$f(x_1,...,x_n)=\sum_{i=1}^{n-1}\left|x_{i+1}-x_i\right| \tag{5.4}$$

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

$$\begin{aligned}&F_i(F_i(x))=F_0(x)=x,\quad i\in\{-1,1\}\\&F_1:0\leftrightarrow 1,2\leftrightarrow 3,.....,254\leftrightarrow 255\\&F_{-1}:-1\leftrightarrow 0,1\leftrightarrow 2,.....,253\leftrightarrow 254,255\leftrightarrow 256\\&F_{-1}(x)=F_1(x+1)-1\qquad \text{her } x \text{ için}\end{aligned} \tag{5.5}$$

Sonra $G$ grubu 3 çesit piksel gruplarda tanımlamaktadır:

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

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

Kullanılmayan gruplar: $G\in U\Leftrightarrow f(F(G))=f(G)$

Her maske için $M$, $F_M(G)=(F_{M(1)}(x_1),.....,F_{M(n)}(x_n))$, $R$ (Regular-Düzenli), $S$ (Single-Tekli), $U$ (Unused-Kullanılmayan) çeşitlerinden birinde tanımlanır. Fridrich işlenmemiş yeni BMP, JPEG ve işlenmiş BMP görüntülerden oluşan büyük veritabanları için 2 istatistiksel tahminleri deneysel olarak ispatlamıştır:

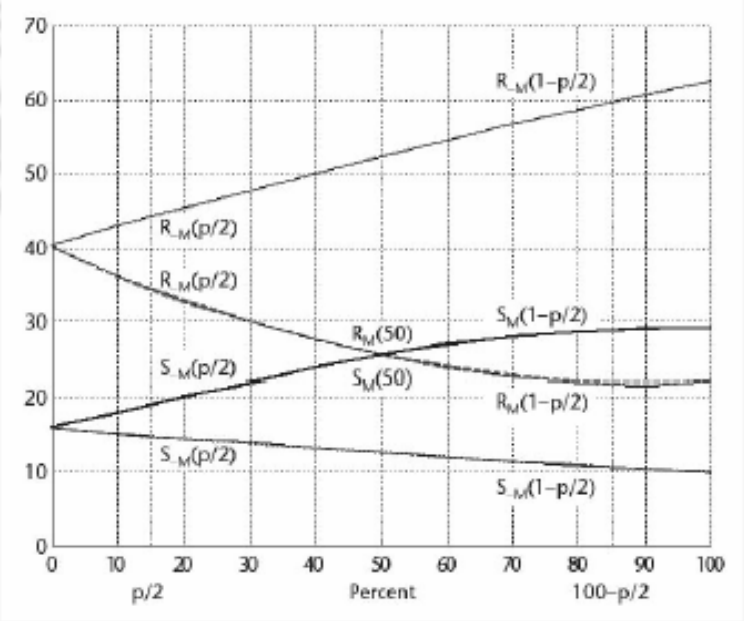
$$R_M+S_M\le 1\quad \text{ve}\quad R_{-M}+S_{-M}\le 1$$

$$R_M\cong R_{-M}\quad \text{ve}\quad S_M\cong S_{-M} \tag{5.6}$$

$$R_M(50)=S_M(50) \tag{5.7}$$

$M$ maskesi $M=[F_0F_1;F_1F_0]$ göstermekte ve $-M$ de $[F_0F_{-1};F_{-1}F_0]$'ı göstermektedir.

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




**Şekil 5.6.** RS Diyagramı

$p$ uzunluklu bir mesaj bir stego görüntü içine gömülmektedir. Mesajın, stego-görüntü piksellerin rasgele bit akımı olduğu farz edilip pikseller kendilerine karşı olanlara gelmektedir. $F_1$ kaydırma işlemi ve $F_{-1}$ kaydırma işlemi bütün piksellere uygulanarak aşağıdaki 4 nokta hesaplanmaktadır:

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

Gizli mesajın uzunluğu için, $p$ (5.6) ve (5.7) varsayımlardan basit bir formül (5.8) üretmek mümkündür.

$$2(d_1+d_0)x^2+(d_{-0}-d_{-1}-d_1-3d_0)x+d_0-d_{-0}=0 \qquad (5.8)$$

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

Mesaj uzunluğu ( $p$ ), $x$ kökünden hesaplanmaktadır:

$$p = x/(x-1/2) \tag{5.9}$$

Eğer bir stego görüntü (5.8).ci denklemin sabit katsayısının 0 olmasını sağlıyorsa (5.6). varsayım bu görüntünün mesaj uzunluğu ( $p$ ) da 0 olarak değerlendirilir.

### 5.3.4. RQP Yöntemi

RQP yöntemi Fridrich tarafından geliştirmiştir [Fridrich vd., 2000]. Bu metod LSB gizlemesi tarafından yaratılan yakın renk çiftlerini analiz etmeye yöneliktir. Öncelikle seçilen resim için yakın renk çiftlerinin tüm renk çiftlerine oranı hesaplanır. Daha sonra bu resim içerisine bir test mesajı gizlenerek oran yeniden hesaplanır. Bu iki oran arasındaki fark büyük ise resminin içinde gizlenmiş bilgi yok demektir. Bu iki oranın birbirine yakın olması resmin içinde gizlenmiş bilgi olduğunu göstermektedir.

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

RQP'ın başka dezavantajı, gri seviyeli görüntülerde uygulanmamasıdır.

### 5.3.5. Görsel Ataklar

Görsel (Visual) ataklarda temel fikir, görüntünün içinde mesaj olan tüm parçalarının görüntüden çıkarılmasıdır. Bu şekilde insan gözü görüntünün içinde bir mesaj olup olmadığını ayırt edebilecektir [Westfeld ve Pfitzmann, 1999]. Görsel atakların amacı görüntüyü insan gözünün değişiklikleri algılayabileceği şekilde gösterebilmektir.

Görsel ataklar daha çok BMP formatındaki dosyalara uygulanabilmektedir. JPEG dosyalar 8x8 piksellik bloklar halinde çalıştığı için bu ataklar sonuç vermemektedir. Görsel ataklarda gizlenen mesaj ya da resmin içindeki bitlerin dağılımı rasgele olmamalıdır.

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

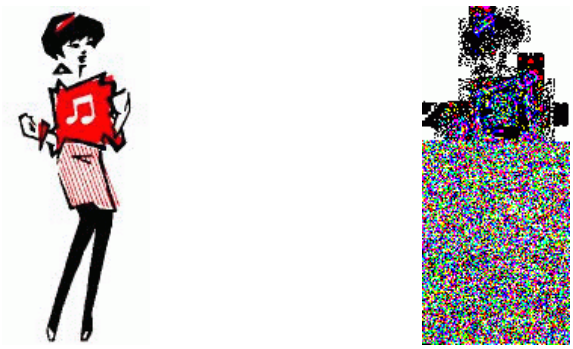


**Şekil 5.7.** Orijinal resim dosyası ve görsel atak sonucu

Şekil 5.8 ve Şekil 5.9 ise sırasıyla içine 1 Kb 5 Kb büyüklüğünde veri gizlenmiş olan görüntü dosyalarını ve görsel atak sonucu elde edilmiş görüntüleri göstermektedir.



**Şekil 5.8.** 1 Kb veri gizlenmiş resim dosyası ve görsel atak sonucu



**Şekil 5.9.** 5 Kb veri gizlenmiş resim dosyası ve görsel atak sonucu

### 5.3.6. JPEG Steganaliz

JPEG Steganaliz JPEG formatındaki dosyalar üzerinde uygulanan steganalitik yöntemlerdir. JPEG dosyalarındaki steganaliz'de iki durum vardır.

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

#### 5.3.6.1. Orijinal ve Bilgi Gizli Resmin Elimizde Olduğu Durum

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

Bu durumu açıklamak için bir örnek ele alalım [El Loco, 2004]. Şekil 5.10'da orijinal resim ve içinde 1,5 KB gizli bilgi bulunan resim verilmektedir. Aslında JPEG dosya formatından dolayı dosya büyüklüklerinde farklılık meydana gelmektedir. Buradan dosyanın içinde bilgi gizli olduğu anlaşılmaktadır. Bir sonraki adım bu bilgiyi elde etmek olacaktır.

| Orijinal resim (16778 byte) | Metin gizlenmiş resim (17344 byte) |
|---|---|
|  |  |



**Şekil 5.10.** Orijinal ve içinde bilgi saklı örnek resim dosyaları

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

| Orijinal resmin ilk 8x8'lik bloğu | İçinde bilgi saklı resmin ilk 8x8'lik bloğu |
|---|---|
| $\begin{bmatrix} D6 & 69 & 13 & 05 & 03 & 15 & F2 & EB \\ FF & 04 & 01 & 00 & FA & FB & F9 & FF \\ 06 & 02 & FE & FF & 00 & 00 & 00 & FF \\ 01 & 03 & 02 & 01 & 01 & FF & 00 & 00 \\ 01 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & FF & FF & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 01 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix}$ | $\begin{bmatrix} D6 & 69 & \underline{12} & 05 & 03 & 15 & \underline{F3} & \underline{EA} \\ \underline{FE} & 04 & \mathbf{01} & \mathbf{00} & FA & FB & \underline{F8} & \underline{FE} \\ 06 & \underline{03} & FE & FF & \mathbf{00} & \mathbf{00} & \mathbf{00} & \underline{FE} \\ \mathbf{01} & \underline{02} & \underline{03} & \mathbf{01} & \mathbf{01} & \underline{FE} & \mathbf{00} & \mathbf{00} \\ \mathbf{01} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} \\ \mathbf{00} & \underline{FE} & FF & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} \\ \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{01} & \mathbf{00} & \mathbf{00} \\ \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} & \mathbf{00} \end{bmatrix}$ |

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

Orijinal resmin nicelendirilmiş DCT katsayı matrisinde 00 ve 01 olan değerlerin olduğu pikseller bit alımında kullanılmamaktadır. O yüzden bilgi gizlenmiş matriste koyu renkle gösterilmiştir ve aşağıdaki hesaplamalarda da göz önüne alınmamaktadırlar.

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

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

- İlk 5 bit kendinden sonraki kaç bitin mesaj uzunluğunu belirlemek için alınacağını belirler. Yani:
  - 00 01 00 01 01 $\Rightarrow$ 01011=11, Bu da ilk beş bitten sonraki 11 bit mesajın uzunluğunu vereceğini belirtir.

- Daha sonraki 11 bit'ten mesajın uzunluğunu bulunur.
  - 01 01 00 00 00 00 01 00 00 00 01⇒ 11000010001=1553
- Sonra gelen 8 bit ise mesajın ilk karakterini vermektedir.
  - 00 01 00 00 01 00 00 01 ⇒ 01001001=73 (Büyük harf I)

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

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

#### 5.3.6.2. Sadece Bilgi Gizli Resmin Elimizde Olduğu Durum

İkinci durumda elimizde sadece şifrelenmiş resim vardır. Bu durumda hangi bloklarda şifrelenmiş metin olduğunu anlamak amacı ile JPEG Uygunluk Esasına Dayanan Steganaliz yapılır [Fridrich ve Goljan, 2002].

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

Stego-görüntüden 8x8 piksellik bloklardaki DCT katsayıların değerlerini analiz ederek JPEG ölçme tablosu elde edilebilmektedir.

Bir görüntüdeki hangi 8x8 piksellik blok JPEG sıkıştırma ile uygunluk göstermiyorsa, bundan gizlenen mesajın uzunluğu ve yerleşimi bulunmaktadır.

Algoritmanın adımları aşağıdaki şekildedir.

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

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

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

**Adım 4:** Her B bloğu için S niceliğini aşağıdaki formülü kullanarak hesapla; $S = \sum_{i=1}^{64} \left| QD^{'}(i) - q_{p(i)}(i) \right|$

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

**ii.** Eğer $S \le 16$ ise her $QD_i$ DCT katsayısı için $Q(i)$'lerin en yakın katlarını hesapla ve $QD_i^{'}$'den uzaklıklarına göre sırala, ve bunları $q_{p(i)}$, p=1,… şeklinde belirt.

**iii.** $S = \sum_{i=1}^{64} \left| QD^{'}(i) - q_{p(i)}(i) \right| \le 16$ eşitsizliğini sağlayan tüm kombinasyonlar için şu eşitliği kontrol et: $B = \left| DCT^{-1}(QD) \right|, \;\; QD(i) = q_{p(i)}(i)$ ise

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

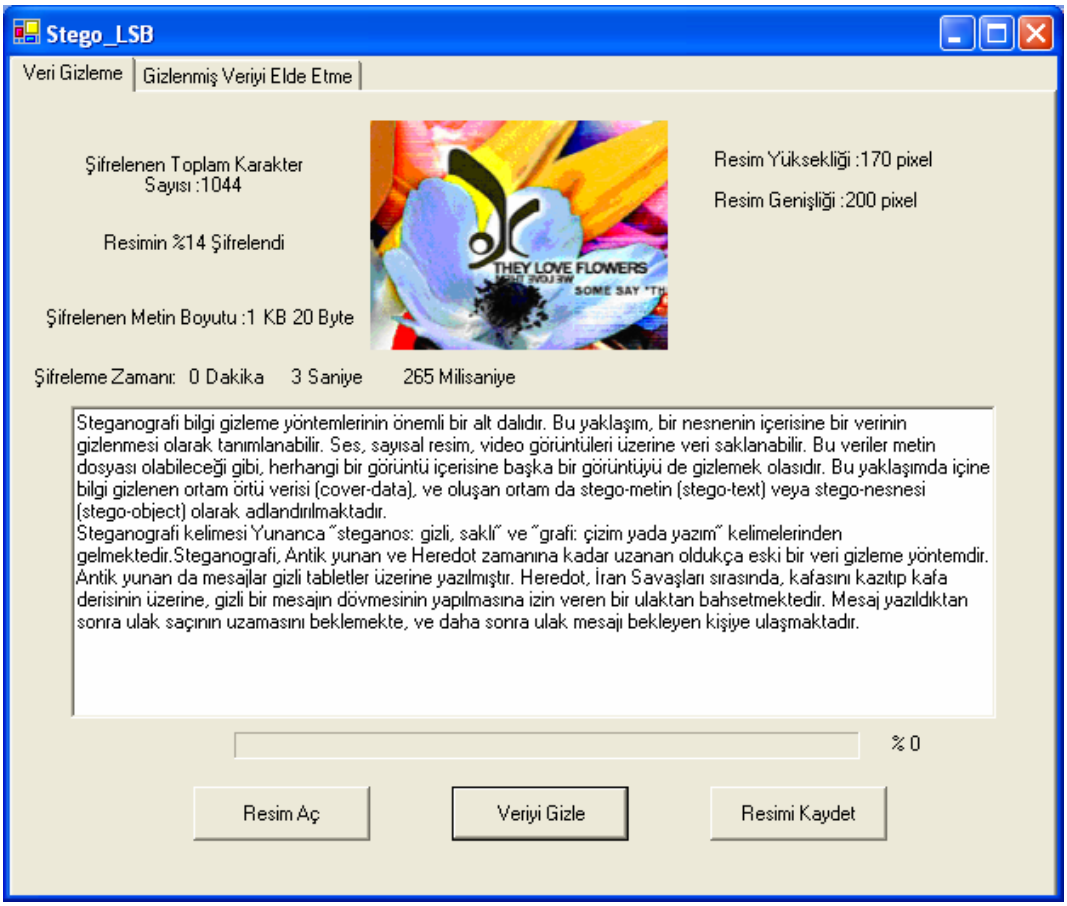
Tüm bloklarının analizinden sonra, uyumsuz JPEG bloğu bulunmaz ise bu resmin içinde gizli bilgi yok demektir. Diğer yandan eğer birkaç tane uyumsuz JPEG bloğu var ise gizli mesaj vardır.

# 6. LSB YÖNTEMİ KULLANILARAK GELİŞTİRİLEN Stego_LSB PROGRAMI VE DEĞERLENDİRİLMESİ

## 6.1. 24-bit Renkli Resimler Üzerinde Sıralı LSB Uygulaması

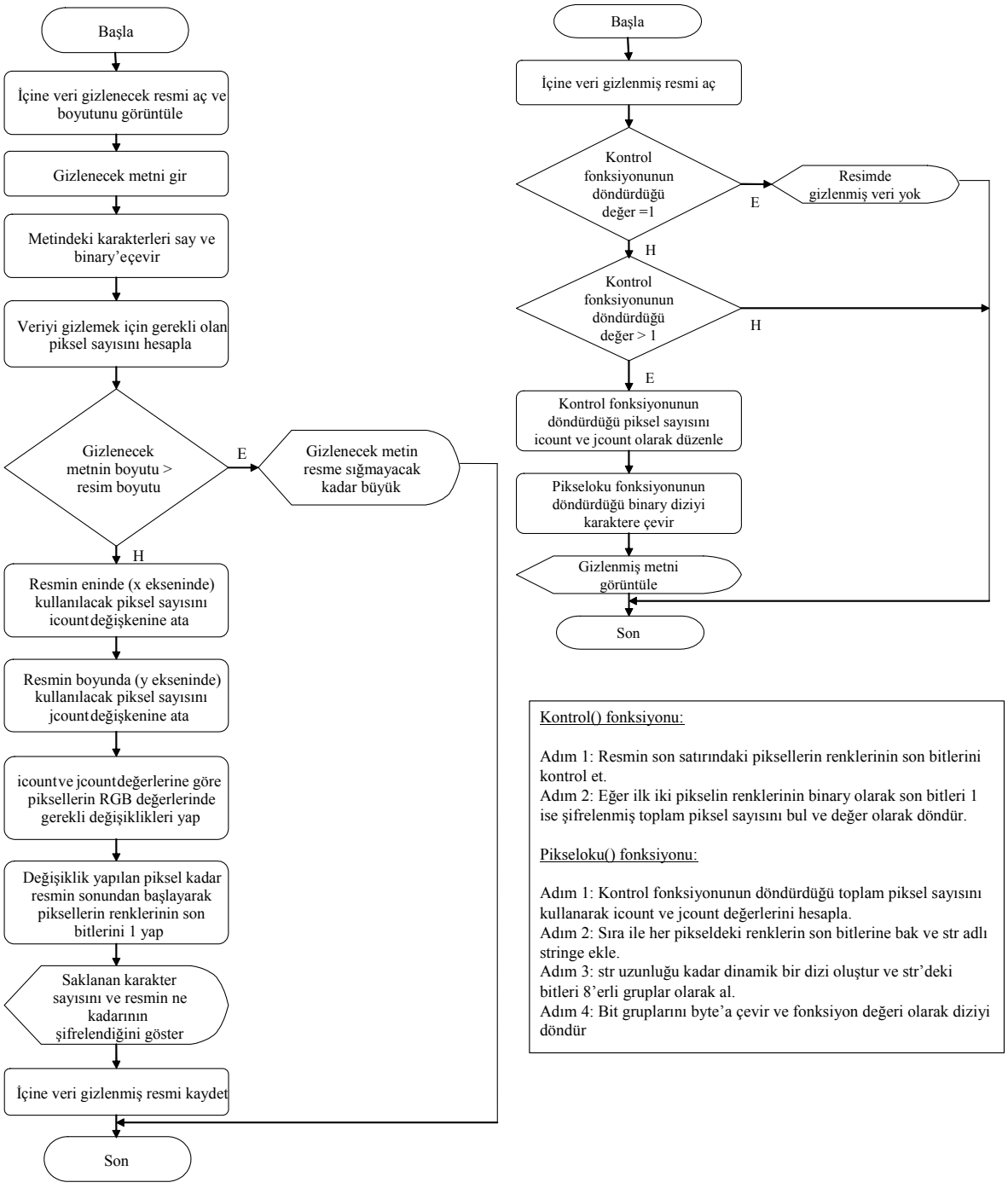
Bu çalışmada 24-bit renkli resimler üzerinde LSB yöntemini kullanarak verileri gizleyen Stego_LSB isimli program [Şahin vd., Haziran 2006] Visual Studio.net kullanılarak geliştirilmiştir. Programın *Veri Gizleme* kısmında öncelikle içine veri gizlenecek resim seçilmekte, daha sonra gizlenmesini istenen metin girilmekte ve şifreleme işlemi yapılmaktadır. *Gizlenmiş Veriyi Elde Etme* kısmında ise içinde veri gizli olan resim dosyası açılmakta ve veriyi elde etme işlemi yapılmaktadır.

Programın örnek bir çalışma penceresi Şekil 6.1'de gösterilmektedir. Burada 1044 byte boyutundaki bir metin resmin içine gizlenmektedir.




**Şekil 6.1.** Programın çalışma penceresi

Şekil 6.2'de, geliştirilen programın veri gizleme-gizlenmiş veriyi elde etme işlemlerinin akış şemaları ve programın kullandığı fonksiyonlar gösterilmektedir.




**Şekil 6.2. (a)** Programın Veri Gizleme işleminin temel akış şeması **(b)** Programın Gizlenmiş Veriyi Elde Etme işleminin temel akış şeması **(c)** Veriyi tekrar elde etme işleminde kullanılan fonksiyonların algoritması

Uygulamada önce 200x170 piksel ve 99,6 KB boyutunda renkli bir resim kullanılmaktadır (resim1.bmp). Şekil 6.3 içine 1 KB (KiloByte) ve 5 KB büyüklüğünde metnin gizlenmesiyle elde edilen resimleri göstermektedir. Şekil 6.2 (a)'da gösterilen algoritma veri gizleme işlemi esnasında resmin kendisini de kullanmaktadır. Bu da gizlenen veri miktarının azalmasına yol açmaktadır. Dolayısıyla resim1.bmp'nin içine boyutundan dolayı en fazla 6,3 KB bilgi saklanabilmektedir. Daha büyük boyuttaki resimlerin içerisine gizlenen veri miktarı doğrusal bir artış göstermektedir.




(a) (b) (c)

**Şekil 6.3. (a)** Orijinal resim (200x170 piksel) **(b)** 1 KB saklanmış resim **(c)** 5 KB saklanmış resim

Diğer bir örnek olan resim2.bmp ise 130x110 piksel - 41,89 KB boyutundadır. Bu resmin içine de aynı büyüklükte metinler gizlenmeye çalışılmıştır, fakat resmin boyutunun küçük olmasından dolayı 5 KB büyüklüğündeki metin içine gizlenememektedir. Resim2.bmp'nin içine saklanabilecek en fazla veri miktarı 2,7 KB'tır. Resim2.bmp'nin içine 1 KB ve 2,7 KB bilgi gizlenerek elde edilen sonuçlar da Şekil 6.4 'te gösterilmektedir.



(a) (b) (c)

**Şekil 6.4. (a)** Orijinal resim (130x110 piksel) **(b)** 1 KB saklanmış resim **(c)** 2,7 KB saklanmış resim

## 6.2. Geliştirilen Uygulamanın Değerlendirilmesi

Geliştiren Stego_LSB isimli program steganografik sistem değerlendirme kriterleri olan taşıyıcıdaki değişim, kapasite ve dayanıklılık açısından incelenmiş ve diğer programlarla karşılaştırılarak o programlara göre hangi seviyede olduğu incelenmiştir.

### 6.2.1. Taşıyıcıdaki Değişim Açısından Değerlendirilmesi

Şekil 6.5'te verilen 175x182 piksellik bir resmin içine 2,7 KB büyüklüğünde bir metin, geliştirilen Stego_LSB ve diğer programlar tarafından saklanmıştır. Bilgi gizlemede kullanılan diğer programlar "Stools-4 [Brown, 1996]", "F5 [Westfeld]", "TurkSteg [Sağıroğlu ve Tunçkanat, 2002]", "Hermetic Stego [Hermetic Systems, 2006]", "InfoStego" [Antiy Labs] ve "Hide and Seek [Maroney]" tir. Taşıyıcıdaki değişimleri ölçmek amacıyla PSNR ve MSE değerleri incelenmiştir. PSNR ve MSE değerleri R, G ve B değerleri için ayrı ayrı hesaplanmaktadır.

Eğer karşılaştırılan dosyalar birebir eşitseler PSNR değeri sonsuz, diğer değerler ise sıfır çıkacaktır. MSE değerlerinin de küçük çıkması beklenmektedir. Yani yüksek PSNR değerine karşılık küçük MSE değerlerinin elde edilmesi resmin çok fazla bozulmadığını göstermektedir.



**Şekil 6.5.** PSNR ölçümleri için kullanılan orijinal resim

**Tablo 6.1.** Elde edilen PSNR ve MSE değerleri

| Kullanılan LSB Programları | | PSNR Değerleri | MSE Değerleri |
|---|---|---|---|
| Stools–4 | R | 60,761400 | 0.054568 |
| | G | 61,164346 | 0.049733 |
| | B | 60,979169 | 0.051900 |
| TurkSteg | R | 57,697705 | 0,110487 |
| | G | 57,810223 | 0,107661 |
| | B | 57,779932 | 0,108414 |
| InfoStego | R | 57.194828 | 0.124050 |
| | G | 57.216868 | 0.123422 |
| | B | 57.080919 | 0.127347 |
| Stego_LSB | R | 54,584353 | 0.226279 |
| | G | 54,587367 | 0.226122 |
| | B | 54,599444 | 0.225495 |
| Hermetic_Stego | R | 50,241110 | 0.615133 |
| | G | 50,246656 | 0.614349 |
| | B | 50,237344 | 0.615667 |
| F5 | R | 31,291584 | 48,297206 |
| | G | 36,392576 | 14,921852 |
| | B | 31,636746 | 44,607284 |
| Hide and Seek | R | 19,769971 | 685.619701 |
| | G | 22,120598 | 399.043783 |
| | B | 20,160764 | 626.619557 |

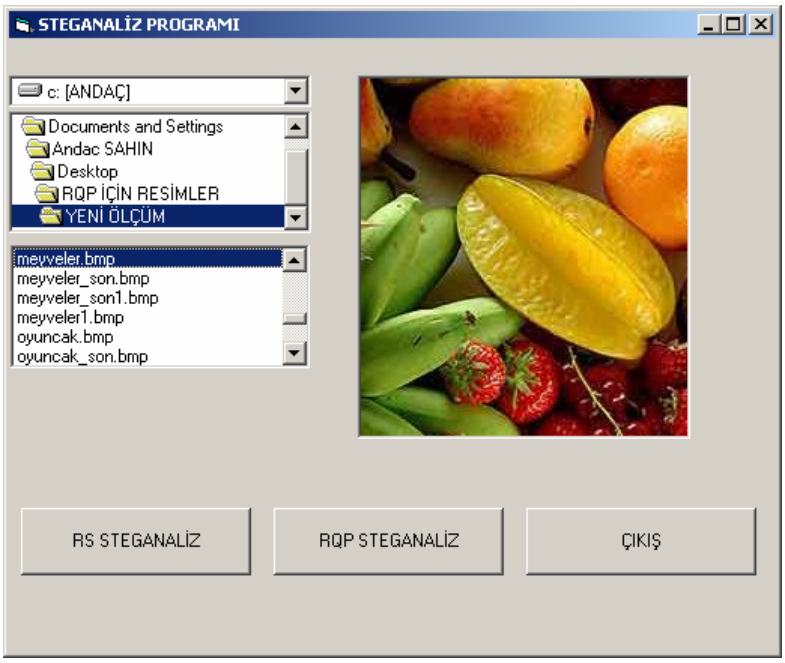
Tablo 6.1'den de görüleceği gibi taşıyıcıdaki değişim açısından geliştirilen program literatürde yaygın olarak kullanılan programların çoğundan daha iyi sonuçlar vermektedir.

### 6.2.2. Kapasite Açısından Değerlendirilmesi

LSB yönteminde kapasite resmin boyutuyla ilgilidir. Bu yüzden aynı büyüklükte resimler için bu yöntemlerin saklayabilecekleri bilgi miktarları eşittir. Eğer yöntem sadece son bite değil sondaki birkaç bite bilgi gizleme işlemi yapıyorsa kapasite değişecektir. Fakat sadece son biti kullanan ve BMP formatındaki dosyalara resim saklayan tüm yöntemlerin saklama kapasiteleri aynıdır.

### 6.2.3. Dayanıklılık Açısından Değerlendirilmesi

Geliştirilen programın dayanıklılık açısından değerlendirilmesi için RS Steganaliz ve RQP Steganaliz yapabilen bir steganaliz uygulaması geliştirilmiştir. Uygulama Microsoft Visual Basic 6.0 ortamında geliştirilmiştir. Geliştirilen uygulamanın örnek bir çalışma penceresi şekil 6.6.'da verilmiştir. Öncelikle 24 bitlik BMP formatındaki resim seçilmekte ve daha sonra istenen steganaliz yöntemi seçilerek resmin içinde bilgi olup olmadığı incelenmektedir.




**Şekil 6.6.** Steganaliz programını örnek bir çalışma penceresi

Ayrıca histogram analizi için Adobe Photoshop 6.0 programı kullanılmıştır.

#### 6.2.3.1. RS Steganaliz Uygulaması

Program 24 bit renkli BMP ya da GIF formatında resmi alıp seçilen maske değerine göre her renk kanalı için ayrı ayrı olmak üzere Düzenli (Regular), Tekil (Singular) ve Kullanılmayan (Unused) grupların sayılarını belirlemektedir. Daha sonra elde edilen değerleri karşılaştırarak bir sonuca varmaktadır. Uygulamanın sözde programı (pseudo code) aşağıda verilmektedir.

**Adım 1:** Resmi seç

**Adım 2:** Maske değerlerini gir.

**Adım 3:** Her renk kanalı için ayrı ayrı uygulanmak üzere;

i. Resmi 4'lü $G$ gruba böl.

ii. $f(G)$ ayırma fonksiyonu değerini hesapla.

iii. Maske ($M$) değerlerine göre uygun kaydırma fonksiyonlarını kullanarak $f(F(G))$ değerini hesapla.

iv. Ayırma ve kaydırma fonksiyonlarından elde edilen değerleri karşılaştırarak Düzenli (R- Regular), Tekil (S- Singular) ve Kullanılmayan (U- Unused) grupların sayılarını belirle.

v. $-M$ için de Adım 3i, 3ii, 3iii ve 3iv'ü tekrarla.

**Adım 4:** Resmin tüm piksellerinin her byte'nın son bitlerini değiştir ve Adım 3'ü tekrarla.

**Adım 5:** Her renk kanalı için orijinal resim ve son bitleri değiştirilmiş resimden elde edilen $R_M$, $S_M$ ve $U_M$ sayıları arasındaki farkı hesapla.

Programın çalışması sonucunda elde edilen fark değerleri 0'a ne kadar yakınsa resmin içinde bilgi yoktur denilebilmektedir.

Öncelikle maske değerlerinin seçiminin ne kadar önemli olduğunu göstermek amacıyla ölçümler yapılmıştır [Şahin vd., Aralık 2006]. Seçilen maske değerlerinin etkinliğini araştırmak için içinde bilgi gizlenmemiş resimler kullanılmıştır. Bu şekilde 0'a en yakın sonuç veren maske değerlerinin en etkin olduğu gözlenebilecektir.

Şekil 6.7'de verilen ve içine bilgi gizlenmemiş olan çeşitli örnek resimler üzerinde değişik maske değerleri denenmiş ve elde edilen sonuçlar tablo 6.2'de verilmiştir. Örnek olarak 4 farklı maske denenmiştir.




(a) sabah.bmp (b) kalp.bmp (c) cicek.bmp (d) scrat.bmp

**Şekil 6.7.** Ölçümler için kullanılan orijinal resimler

**Tablo 6.2.** (1,0,-1,1), (1,0,1,-1), (0,1,1,-1) ve (0,-1,1,-1) maske değerleri kullanılarak elde edilen Düzenli (Regular-R), Tekil (Singular-S) ve Kullanılmayan (Unused-U) gruplar arası fark değerleri. Değerler R (Red-Kırmızı), G (Gren-Yeşil) ve B (Blue-Mavi) renk kanalları için ayrı ayrı hesaplanmıştır.

**(a)** sabah.bmp için elde edilen değerler

| Maske değerleri→ | | (1,0,-1,1) | (1,0,1,-1) | (0,1,1,-1) | (0,-1,1,-1) |
|---|---|---|---|---|---|
| R (Kırmızı) renk kanalı için | R | 10 | 21 | 35 | 12 |
| | S | 9 | 32 | 35 | 12 |
| | U | 1 | 53 | 0 | 0 |
| G (Yeşil) renk kanalı için | R | 9 | 15 | 79 | 4 |
| | S | 15 | 29 | 79 | 4 |
| | U | 24 | 44 | 0 | 0 |
| B (Mavi) renk kanalı için | R | 15 | 19 | 17 | 7 |
| | S | 18 | 20 | 17 | 7 |
| | U | 33 | 39 | 0 | 0 |

**(b)** kalp.bmp için elde edilen değerler

| Maske değerleri→ | | (1,0,-1,1) | (1,0,1,-1) | (0,1,1,-1) | (0,-1,1,-1) |
|---|---|---|---|---|---|
| R (Kırmızı) renk kanalı için | R | 97 | 119 | 1132 | 23 |
| | S | 299 | 299 | 1132 | 23 |
| | U | 202 | 180 | 0 | 0 |
| G (Yeşil) renk kanalı için | R | 128 | 166 | 1147 | 7 |
| | S | 335 | 257 | 1147 | 7 |
| | U | 207 | 91 | 0 | 0 |
| B (Mavi) renk kanalı için | R | 140 | 179 | 949 | 12 |
| | S | 306 | 195 | 949 | 12 |
| | U | 166 | 16 | 0 | 0 |

**(c)** cicek.bmp için elde edilen değerler

| Maske değerleri→ | | (1,0,-1,1) | (1,0,1,-1) | (0,1,1,-1) | (0,-1,1,-1) |
|---|---|---|---|---|---|
| R (Kırmızı) renk kanalı için | R | 3 | 115 | 340 | 13 |
| | S | 21 | 189 | 340 | 13 |
| | U | 18 | 304 | 0 | 0 |
| G (Yeşil) renk kanalı için | R | 36 | 139 | 546 | 5 |
| | S | 77 | 228 | 546 | 5 |
| | U | 113 | 367 | 0 | 0 |
| B (Mavi) renk kanalı için | R | 71 | 125 | 280 | 29 |
| | S | 54 | 160 | 280 | 29 |
| | U | 125 | 285 | 0 | 0 |

**(d)** scrat.bmp için elde edilen değerler

| Maske değerleri→ | | (1,0,-1,1) | (1,0,1,-1) | (0,1,1,-1) | (0,-1,1,-1) |
|---|---|---|---|---|---|
| R (Kırmızı) renk kanalı için | R | 45 | 37 | 351 | 7 |
| | S | 165 | 163 | 351 | 7 |
| | U | 210 | 200 | 0 | 0 |
| G (Yeşil) renk kanalı için | R | 32 | 51 | 366 | 36 |
| | S | 66 | 95 | 366 | 36 |
| | U | 98 | 146 | 0 | 0 |
| B (Mavi) renk kanalı için | R | 165 | 172 | 343 | 9 |
| | S | 209 | 232 | 343 | 9 |
| | U | 374 | 404 | 0 | 0 |

Maske değerlerinin seçimi oldukça önemlidir. Örnek olarak seçilen 4 resim üzerinde (1,0,-1,1), (1,0,1,-1), (0,1,1,-1) ve (0,-1,1,-1) maske değerleri kullanılarak RS steganaliz uygulanmıştır.

Tabloda bulunan fark değerlerinin 0'a yakın olması resmin içinde bilgi olmadığını ve doğru bir maske seçildiğini göstermektedir. Tablodan da görülebileceği gibi fark değeri en az olan maske, tüm resimler için (0,-1,1,-1)'dir. Aradaki farkın en fazla olduğu maske değerinin ise (0,1,1,-1) olduğu görülmektedir.

Daha sonra geliştirilen Stego_LSB isimli programın güvenilirliğini incelemek amacıyla, diğer LSB programlarıyla bilgi gizlenmiş resimler RS Steganaliz açısından karşılaştırılmıştır. Bunun için şekil 6.8'deki 260x180 piksel boyutundaki resmin içerisine 5,4 KB boyutundaki metin "Stools–4", "F5", "TurkSteg", "Hermetic Stego", "InfoStego" ve "Stego_LSB" programları tarafından gizlenmiştir.



**Şekil 6.8.** RS steganaliz için kullanılan orijinal resim

Seçilen değişik maske değerlerine göre elde edilen sonuçlar tablo 6.3, tablo 6.4 ve tablo 6.5'te gösterilmiştir.

**Tablo 6.3.** $M = (0,1,0,-1)$ olduğu durumda elde edilen sonuçlar.

| | | Orijinal resim | STools_4 | F5 | TurkSteg | InfoStego | Hermetic Stego | Stego_LSB |
|---|---|---|---|---|---|---|---|---|
| **R** | **R** | 1 | 2 | 38 | 29 | 0 | 5 | 1 |
| | **S** | 1 | 2 | 38 | 29 | 0 | 5 | 1 |
| | **U** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **G** | **R** | 3 | 13 | 1 | 8 | 7 | 14 | 6 |
| | **S** | 3 | 13 | 1 | 8 | 7 | 14 | 6 |
| | **U** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **B** | **R** | 21 | 23 | 40 | 16 | 21 | 8 | 11 |
| | **S** | 21 | 23 | 40 | 16 | 21 | 8 | 11 |
| | **U** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Tablo 6.4.** $M = (1,-1,1,0)$ olduğu durumda elde edilen sonuçlar.

| | | Orijinal resim | STools_4 | F5 | TurkSteg | InfoStego | Hermetic Stego | Stego_LSB |
|---|---|---|---|---|---|---|---|---|
| **R** | **R** | 7 | 6 | 12 | 10 | 7 | 1 | 5 |
| | **S** | 7 | 6 | 12 | 10 | 7 | 1 | 5 |
| | **U** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **G** | **R** | 1 | 0 | 5 | 3 | 2 | 5 | 1 |
| | **S** | 1 | 0 | 5 | 3 | 2 | 5 | 1 |
| | **U** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **B** | **R** | 11 | 11 | 11 | 10 | 8 | 8 | 7 |
| | **S** | 11 | 11 | 11 | 10 | 8 | 8 | 7 |
| | **U** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Tablo 6.5.** $M = (-1,0,-1,1)$ olduğu durumda elde edilen sonuçlar.

| | | Orijinal resim | STools_4 | F5 | TurkSteg | InfoStego | Hermetic Stego | Stego_LSB |
|---|---|---|---|---|---|---|---|---|
| **R** | **R** | 44 | 29 | 3 | 18 | 30 | 3 | 20 |
| | **S** | 55 | 45 | 63 | 32 | 47 | 11 | 32 |
| | **U** | 99 | 74 | 66 | 50 | 77 | 8 | 52 |
| **G** | **R** | 34 | 37 | 32 | 10 | 40 | 48 | 26 |
| | **S** | 53 | 67 | 66 | 24 | 64 | 29 | 35 |
| | **U** | 87 | 104 | 98 | 34 | 104 | 77 | 61 |
| **B** | **R** | 41 | 43 | 51 | 31 | 42 | 19 | 23 |
| | **S** | 52 | 46 | 106 | 43 | 54 | 20 | 38 |
| | **U** | 93 | 89 | 154 | 74 | 96 | 39 | 61 |

Tablolardaki değerler $R_M$ ile $R_{-M}$ , $S_M$ ile $S_{-M}$ ve $U_M$ ile $U_{-M}$ arasındaki farkları göstermektedir. Bu farkların 0'a yakın çıkması mesaj olmadığını söylemektedir. Tablolardan da görülebileceği gibi maske değerlerinin seçimi de oldukça önemlidir. Yanlış bir maske seçimiyle hatalı sonuçlar elde edilebilir. Tablo 6.5 böyle bir durumu göstermektedir.

Tablo 6.3 ve tablo 6.4'te seçilen maske değerlerine göre orijinal resmin fark değerlerinin 0'a yakın olduğu görülmektedir. Şifrelenmiş resimlerde de farklı maske değerleri durumlarında da Stego_LSB programımızın oldukça iyi sonuçlar verdiği görülmektedir.

#### 6.3.3.2. RQP Steganaliz Uygulaması

Program 24 bit renkli resimler üzerinde çalışmaktadır. Öncelikle seçilen resim için yakın renk çiftlerinin tüm renk çiftlerine oranı hesaplanmaktadır. Daha sonra bu resim içerisine bir test mesajı gizlenerek oran yeniden hesaplanır. Bu iki oran arasındaki farkın büyük olması resminin içinde gizlenmiş bilgi olmadığını göstermektedir. Bu iki oranın birbirine yakın olması ise resmin içinde gizlenmiş bilgi olduğunu göstermektedir.

Programın sahte kodu aşağıda verilmiştir.

**Adım 1:** Resmi seç.

**Adım 2:** Yakın renk çiftlerinin sayısını hesapla (renk çiftleri arasındaki fark 3'ten küçük olanlar yakın renk çifti olarak seçilmiştir.)

**Adım 3:** Yakın renk çiftlerinin tüm renk çiftlerine oranını hesapla ve $O1$ olarak belirle.

**Adım 4:** Seçilen resmin içine bir test mesajı gizle ve oranı tekrar hesaplayıp $O2$ olarak belirle.

**Adım 5:** $O1$ ile $O2$ arasındaki farkı hesapla.

Programın çalışmasını incelemek amacıyla örnek olarak 10 adet resim seçilmiş ve Şekil 6.9'da gösterilmiştir. Öncelikle içinde bilgi gizli olmayan resimlere RQP steganaliz uygulanmış ve elde edilen sonuçlar Tablo 6.6'da verilmiştir. Daha sonra aynı resimlerin içerisine bir metin gizlenmiştir ve tekrar RQP Steganaliz uygulanmıştır. Bunun sonucunda elde edile değerler ise Tablo 6.7'de gösterilmiştir.



(a) ataturk.bmp 400x300 piksel | (b) bahce.bmp 335x192 piksel | (c) balik.bmp 379x253 piksel | (d) cicek.bmp 312x223 piksel | (e) kalp.bmp 313x292 piksel



(f) kartal.bmp 269x249 piksel | (g) meyve.bmp 217x238 piksel | (h) oyuncak.bmp 240x192 piksel | (ı) resim.bmp 336x240 piksel | (i) scrat.bmp 292x308 piksel

**Şekil 6.9.** RQP Steganaliz için kullanılan örnek resimler

**Tablo 6.6.** İçine bilgi gizlenmemiş resimlere uygulanan RQP steganaliz sonuçları

| | *O*1 | *O*2 | *Fark* |
|---|---|---|---|
| ataturk.bmp | 0,30413 | 0,28312 | 0,02101 |
| bahce.bmp | 0,10332 | 0,09098 | 0,01235 |
| balik.bmp | 0,23569 | 0,22222 | 0,01347 |
| cicek.bmp | 0,35365 | 0,32045 | 0,03320 |
| kalp.bmp | 0,91485 | 0,91125 | 0,00360 |
| kartal.bmp | 0,68488 | 0,65845 | 0,02643 |
| meyve.bmp | 0,31941 | 0,28589 | 0,03352 |
| oyuncak.bmp | 0,12483 | 0,11404 | 0,01079 |
| resim.bmp | 0,39259 | 0,37610 | 0,01649 |
| scrat.bmp | 0,40881 | 0,38609 | 0,02272 |

**Tablo 6.7.** İçinde bilgi gizli olan resimlere uygulanan RQP steganaliz sonuçları

| | *O*1 | *O*2 | *Fark* |
|---|---|---|---|
| ataturk.bmp | 0,28615 | 0,28312 | 0,00302 |
| bahce.bmp | 0,09311 | 0,09098 | 0,00213 |
| balik.bmp | 0,22408 | 0,22222 | 0,00186 |
| cicek.bmp | 0,32547 | 0,32045 | 0,00502 |
| kalp.bmp | 0,91182 | 0,91125 | 0,00057 |
| kartal.bmp | 0,66287 | 0,65845 | 0,00442 |
| meyve.bmp | 0,29172 | 0,28589 | 0,00583 |
| oyuncak.bmp | 0,11612 | 0,11404 | 0,00208 |
| resim.bmp | 0,37729 | 0,37610 | 0,00119 |
| scrat.bmp | 0,38988 | 0,38609 | 0,00379 |

Tablo değerlerinden de görüleceği gibi içinde bilgi gizli olmayan resim dosyalarına uygulanan RQP steganaliz sonucunda fark değerlerinin yüzde seviyesinde olduğu görülmektedir. İçinde bilgi gizli olan dosyalarda ise bu fark binde seviyesine düşmektedir. Bu nedenle programın çalışması sonucunda elde edilen değerler binde seviyesinde ise resim içinde bilgi gizlenmiştir denilebilir.

Stego_LSB programının ne kadar etkin olduğunu anlamak amacıyla da Şekil 6.8'deki orijinal resim içerisine "Stools–4", "F5", "TurkSteg", "Hermetic Stego", "InfoStego" ve "Stego_LSB" programları tarafından bir metin gizlenmiş ve ölçümler yapılmıştır. Elde edilen sonuçlar tablo 6.8.'de verilmiştir.

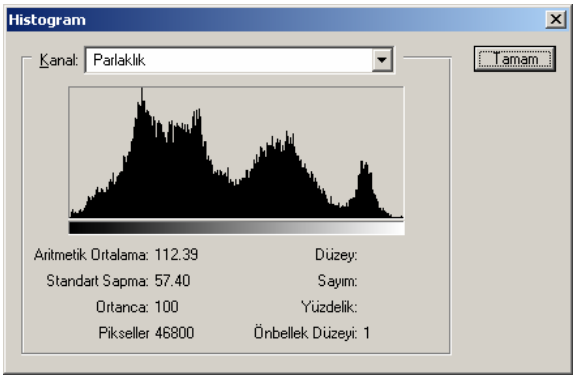
**Tablo 6.8.** Şekil 6.8'deki resim üzerinde uygulanan RQP Analizi

| | *O*1 | *O*2 | *Fark* |
|---|---|---|---|
| S-Tools4 | 0,21936 | 0,19483 | 0,02453 |
| TurkSteg | 0,21263 | 0,19483 | 0,01780 |
| InfoStego | 0,21327 | 0,19962 | 0,01365 |
| F5 | 0,24282 | 0,22955 | 0,01327 |
| Stego_LSB | 0,19538 | 0,19301 | 0,00237 |
| Hermetic Stego | 0,19209 | 0,19376 | 0,00167 |

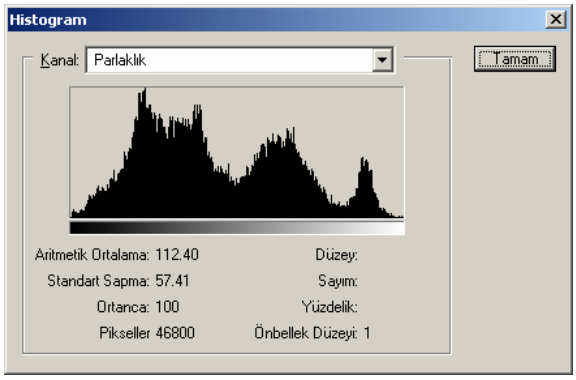
Burada fark değeri 0'a ne kadar yakınsa resmin içinde bilgi var demektir. Tablo 6.9.'dan elde edilen değerlere göre Stego_LBS programı ile içine bilgi saklanan resim dosyası RQP steganaliz ile incelendiğinde gizli bilginin varlığı daha kolay fark edilmektedir. Bu nedenle Stego_LSB programı RQP steganaliz açısından, Hermetic Stego hariç, kullanılan diğer programlardan RQP steganalizde biraz daha kötü bir performans sergilemektedir.

#### 6.3.3.3. Histogram Analizi

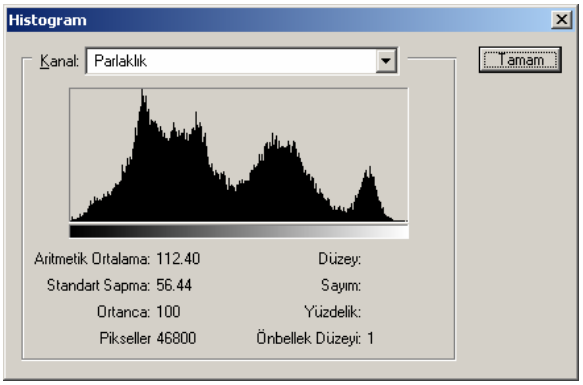
Histogram analizi için Adobe Photoshop 6.0 programı kullanılmıştır. Bu amaçla şekil 6.8'deki 260x180 piksel boyutundaki resmin içerisine 5,4 KB boyutundaki metin "Stools-4", "F5", "Turk_Steg", "Hermetic Stego", "InfoStego" ve "Stego_LSB" programları tarafından gizlenmiştir. Elde edilen resimlerin parlaklık değerlerinin (Y) histogramları çıkartılmıştır. Şekil 6.10'da orijinal resmin ve kullanılan bilgi gizleme programları sonucunda elde edilen resimlerin histogramları gösterilmektedir.



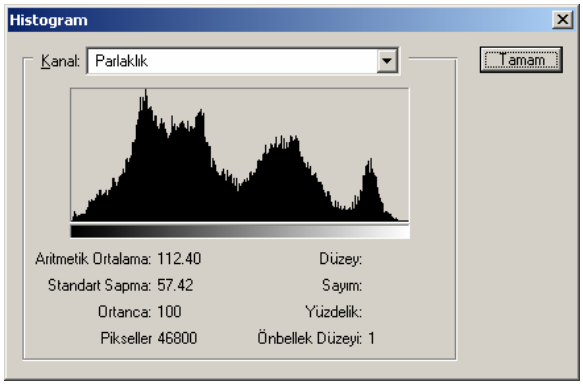

(a) Orijinal resmin histogramı



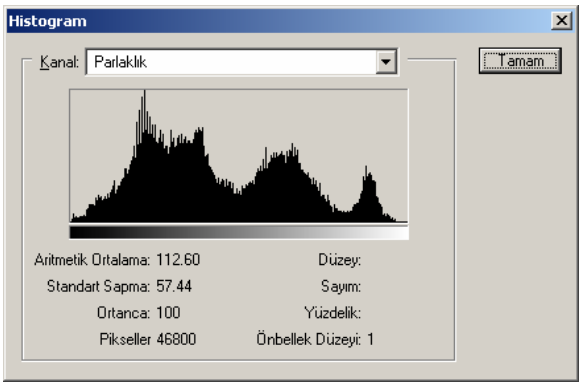

(b) InfoStego programı ile bilgi gizlenen resmin histogramı



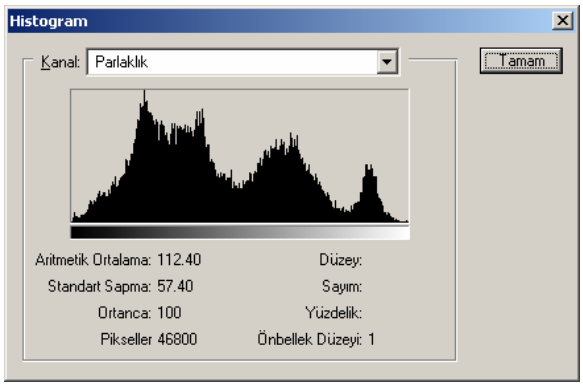

(c) F5 algoritması ile bilgi gizlenen resmin histogramı




(d) Hermetic Stego programı ile bilgi gizlenen resmin histogramı




(e) Stego_LSB programı ile bilgi gizlenen resmin histogramı




(f) STools-4 programı ile bilgi gizlenen resmin histogramı

**Şekil 6.10.** Orijinal ve bilgi gizlenmiş resimlerin histogramları

Histogram analizi orijinal resmin elimizde olmadığı durumlarda pek işe yaramamaktadır. Çünkü dağılımın orijinal resimden ne kadar farklı olduğunu bilinmemektedir. Farklı programlar tarafından bilgi gizlenen resmin histogramları çok fazla değişiklik göstermemektedir. Programların histogram analizine karşı dayanıklılıkları yaklaşık olarak aynıdır.

# 7. SONUÇLAR

Bu tezde sayısal görüntü dosyalarına bilgi gizlemek için kullanılan steganografik yöntemler ve bu yöntemlerin güvenilirlikleri incelenmiştir. Bir steganografik sistem sağlamlık ve şeffaflık ölçütlerini sağlamalıdır. Şeffaflık saklanan verinin tespit edilememesi ve fark edilememesini ifade ederken sağlamlık saklanan verinin çıkartma işleminde düzgün bir şekilde geri getirilmesini anlatmaktadır. Bir steganografik sistemin değerlendirilmesi içinse taşıyıcıyı ne kadar değiştirdiğine, saklayabileceği bilgi miktarına ve steganalitik saldırılara karşı ne kadar dayanıklı olduğunun incelenmesi gerekmektedir.

Taşıyıcıdaki değişimleri ölçebilmek amacıyla PSNR ve MSE değerleri hesaplanmaktadır. Burada yüksek PSNR değerlerine karşılık düşük MSE değerlerinin elde edilmesi resmin fazla bozulmadığını yani taşıyıcıdaki değişimin fazla olmadığını göstermektedir. JPEG ve GIF formatındaki resim dosyaları özelliklerinden dolayı taşıyıcıdaki değişime daha duyarlıdır. Taşıyıcıdaki değişim açısından BMP dosyalar daha iyi sonuçlar vermektedir.

Kapasite açısından bakıldığında ise BMP ve GIF formatındaki resimlerin daha fazla bilgiyi saklayabilmelerinden dolayı tercih edildiği görülmüştür. JPEG formatındaki resimler dosya yapılarından dolayı her 8x8 piksellik bloklara 1 byte saklayabilmektedirler. Bu yüzden saklama kapasitelerinin düşük olduğu söylenebilir.

Dayanıklılık ise çeşitli steganalitik saldırılara karşı ne kadar başarılı olduğu ile değerlendirilmektedir. Her steganografik yöntem için ayrı steganaliz yöntemleri geliştirilmiştir. Bir yöntem için çok iyi sonuçlar veren bir steganaliz yöntemi bir diğeri için doğru sonuç vermeyebilir. Genelde JPEG formatındaki resimler steganalitik saldırılara karşı daha başarılıdırlar.

Bu çalışmada sıralı LSB ekleme yöntemine göre çalışan bir uygulama geliştirilmiş ve yukarıdaki ölçütlere göre incelenmiştir.

Taşıyıcıdaki değişime göre incelendiğinde Stego_LSB programının literatürde yaygın olarak kullanılan "Hermetic Stego", "F5" ve "Hide and Seek" gibi önemli programlardan daha iyi sonuçlar verdiği görülmüştür.

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

Dayanıklılığı ölçmek amacıyla RQP ve RS steganaliz yapabilen bir uygulama geliştirilmiş ve ayrıca Adobe Photoshop programı kullanılarak histogram analizi yapılmıştır. Ölçümler için 260x180 piksel boyutunda bir örnek resim seçilmiş ve içerisine 5,4 KB boyutundaki metin "Stools–4", "F5", "TurkSteg", "Hermetic Stego", "InfoStego" ve "Stego_LSB" programları tarafından gizlenmiştir.

RS Steganaliz sonucunda elde edilen fark değerlerinin 0'a yakın çıkması resmin içinde bilgi olmadığını veya resmin içinde bilgi varsa yöntemin RS Steganaliz'e karşı dayanıklı olduğunu göstermektedir. Yapılan ölçümler sonucunda farklı maske değerleri kullanılsa bile Stego_LSB programının diğer programlara göre genelde 0'a yakın değerler ürettiği gözlenmiştir. Bu durumda Stego_LSB programıyla resmin içine gizlenen bilginin RS Steganaliz yöntemiyle sezilme olasılığı düşüktür denilebilir.

RQP Steganaliz açısından bakıldığında ise $O_1$ ile $O_2$ arasındaki fark değerlerinin yüzde seviyesinde çıkması resmin içinde bilgi olmadığını; binde seviyesinde çıkması resmin içinde bilgi olduğunu göstermektedir. Yapılan ölçümler neticesinde Stego_LSB programı kullanılarak resim içine bilgi saklandığında RQP Steganalizle sezilme olasılığının daha yüksek olduğu söylenebilir. Fakat bu durum resmin renk dağılımıyla da ilgilidir.

Histogram analizinde ise elimizde orijinal resmin histogramı yoksa kesin bir sonuca varmamız oldukça zordur. İçine aynı büyüklükte farklı programlar kullanılarak veri gizlenen resimlerin histogramları arasında çok büyük farklılıklar olmadığı gözlenmiştir.

Günümüzde birçok steganografik yöntem bulunmaktadır. Bir steganografik yöntem değerlendirilirken dayanıklılık - kapasite ve taşıyıcıdaki değişim - kapasite arasında ikilemler sözkonusudur. Kapasite arttıkça dayanıklılık azalacaktır. Yine aynı şekilde kapasite miktarı arttıkça taşıyıcı ortamdaki değişimler artacaktır. Taşıyıcıdaki değişim, dayanıklılık ve kapasite özelliklerinden hangisi bizim için daha önemliyse bu duruma göre daha uygun olan yöntem veri gizleme işlemi için kullanılmalıdır.

Steganografik yöntemler şifreleme yöntemleri ile birlikte kullanılarak daha güvenli bir sistem oluşturulabilir. Saklanacak verinin miktarı arttırılmak istenirse de gizlenecek verinin gizleme işleminden önce sıkıştırılmasıyla bu sağlanabilmektedir.

Bir resmin içinde gizli bilgi olduğunun anlaşılmasından sonra yapılacak işlem bu verinin elde edilmesidir. Fakat bunun için bilgi gizlemede kullanılan steganografik yöntemin bilinmesi gerekmektedir. Resmin içindeki bilginin elde edilmesi uğraş gerektiren ve zaman alan bir süreçtir. İnternet üzerinden hergün milyonlarca resim ya da video dosyası gönderildiği düşünülürse gizli bilgilerin sezilmesi bile oldukça zordur. Steganografinin kötü amaçlar için kullanılması durumunda insanlık açısından kötü sonuçlar ortaya çıkabilmektedir. Steganografik yöntemlerin çeşitliliği ve her steganaliz yönteminin gizli verileri yakalayamaması dolayısıyla kötü amaçlı kişiler bu yöntemleri tercih etmeye başlamışlardır. Bu nedenle steganografi ve steganaliz yöntemleri gelişmeye ve ilerlemeye oldukça açık bir konudur.

# KAYNAKLAR


1. Alwan R.H., Kadhim F.J., Al-Taani A.T., "Data Embedding Based on Better Use of Bits in Image Pixels", International Journal of Signal Processing Volume 2, Number 2, ISSN 1304-4494, 2005.
2. Amin M. M., Salleh M., Ibrahim S., Katmin M. R., "Steganography: Random LSB Insertion Using Discrete Logarithm," Proceedings of 3rd International Conference on Information Technology in Asia (CTA03), pp. 234–238, 2003.
3. Anderson R.J., ed., Information Hiding: First International Workshop, vol 1174 of Lecture Notes in Computer Science, Isaac Newton Institute, Cambridge, England, Springer-Verlag, Berlin, Germany, ISBN 3-540-61996-8, May 1996.
4. Anderson R.J, Petitcolas F.A.P., "On the Limits of Steganography", IEEE Journal of Selected Areas in Communications, 16(4):474-481, Special Issue on Copyright & Privacy Protection. ISSN 0733–8716, May 1998.
5. Antiy Labs, InfoStego, http://www.antiy.net/infostego/
6. Aura T., "Invisible Communication", EET 1995, Technical Report, Helsinki University of Technology, Finland, November 1995.
7. Atıcı M.A., "Steganografik Yaklaşımlar ve Uygulamaları", Gazi Üniversitesi, Bilgisayar Mühendisliği Bölümü Yüksek Lisans Semineri, 2005.
8. Barni M., Bartolini F., Cappellini V., Piva A., "Robust Watermarking of Stil Images for Copyright Protection", in Proceedings of DSP'97, International Conference on Digital Signal Processing, Santorini, Greece, pp.499-502, 2-4 July 1997.
9. Bender W., Gruhl D., Morimoto N., Lu A., "Techniques for data hiding", IBM Systems Journal, vol. 35, NOS 3&4, 1996.
10. Biryukov A., "Methods of Cryptanalysis", PHd Thesis, 1999.

11. Brown A., “S-Tools for Windows”, ftp://idea.sec.dsi.unimi.it/pub/security/ crypt/code/s-tools4.zip, 1996.

12. Cabeen K., Gent P., “Image Compression and the Discrete Cosine Transform”, http://online.redwoods.cc.ca.us/instruct/darnold/laproj/Fall98/PKen/dct.pdf,1998

13. Chaum D., “Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms”, Communications of the ACM, vol. 24, no. 2, pp. 84-88, February 1981.

14. CompuServe, “Graphics Interchange Format(sm), Version 89a”, CompuServe Incorporated, Columbus, Ohio, 1990.

15. Cox I.J., Kilian J, Leighton T., Shamoon T., “A Secure, Robust Watermark for Multimedia”, Proc. First Int’l Workshop Information Hiding, Lecture Notes in Computer Science No. 1, 174, Springer-Verlag, Berlin, 1996, pp. 185-206.

16. Deogol, http://wandership.ca/projects/deogol/intro.html#deogol

17. Einstein A., “On the electrodynamics of moving bodies”, Ann. Phys., Lpz. 17, pp. 891-921, 1905.

18. El Loco G., “Extracting data embedded with JSteg”, http://www.guillermito2.net/ stegano/jsteg/index.html, 2004.

19. Erkin Z., Örencik B., “Steganografik Kütüphane“, Ağ ve Bilgi Güvenliği Ulusal Semp. (ABG 2005) Bildiriler Kitabı, ISBN 975-395-885-4, s. 98-102, İstanbul, 9-11 Haziran, 2005.

20. EzStego, http://www.fqa.com/romana/

21. Farid H., “Steganography: the art and mathematics of hiding information”, Teaching Notes, www.cs.dartmouth.edu/farid/teaching/cs4/summer.03/notes/ steg.pdf, 2003.

22. FIPS 46-3, Data Encryption Standard, Federal Information Processing Standard (FIPS), Publication 46-3, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., October 25, 1999.

23. Fridrich J., Goljan M., “Practical Steganalysis of Digital Images – State of the Art”, Proc. SPIE Photonics West, Vol. 4675, Electronic Imaging 2002, Security and Watermarking of Multimedia Contents, pp. 1-13, San Jose, California, January 2002.

24. Fridrich J., Du R., Meng L., “Steganalysis of LSB Encoding in Color Images”, Proceedings IEEE International Conference on Multimedia and Expo, New York City, NY, July 30–August 2, 2000.

25. Fridrich J., Goljan M., Hogea D., “Steganalysis of JPEG Images: Breaking the F5 Algorithm”, 5th Information Hiding Workshop, Noordwijkerhout, The Netherlands, pp. 310-323, 7–9 October 2002.

26. Gifshuffle, http://www.darkside.com.au/gifshuffle/

27. Goldschlag D.M., Reed M.G., Syverson P.F., “Hiding Routing Information”, in Information Hiding: First International Workshop, Proceedings, vol 1174 of Lecture Notes in Computer Science, Springer, pp. 84-88, 1996.

28. Gruhl D., Lu A., Bender W., “Echo Hiding”, in Proceeding of First International Workshop, Springer, Cambridge, UK, May-June, 1996.

29. Hamilton, E., 1992, “JPEG File Interchange Format Version 1.02”.

30. Hartung F., Kutter M., “Multimedia watermarking techniques,” Proc. of the IEEE, vol. 87, pp. 1079–1107, July 1999.

31. Heredotus, “The Histories”, First Published in Greek 430 B.C.E., Translated by Selincourt A., Penguin Classics, Hammondsworth, 1971.

32. Hermetic Systems, “Hermetic Stego”, http://www.hermetic.ch/hst/intro.htm, Program: http://www.hermetic.ch/hst/hst_setup.zip, 2006.

33. http://wwwrn.inf.tu-dresden.de/~westfeld/f5.html

34. http://www.stegoarchive.com/

35. Huitsing P., http://www.personal.utulsa.edu/~peter-huitsing/steganography.html.

36. Johnson N.F., “Steganography”, online paper, http://www.jjtc.com/stegdoc/ stegdoc.html, 1996.

37. Johnson N.F., Duric Z., Jajodia S., “Information Hiding: Steganography and Watermarking - Attacks and Countermeasures”, Kluwer Academic Publishers, ISBN: 0-79237-204-2, 2000.

38. Johnson N.F, Jajodia S., “Exploring steganography: Seeing the Unseen”, Computer, 31, no 2:26-34, February 1998.

39. Johnson N. F., Jajodia S., “Steganalysis of Images Created Using Current Steganography Software”, Second Information Hiding Workshop held in Portland, Oregon, USA, April 15-17, 1998. Proceedings LNCS 1525, 273-289, Springer-Verlag, 1998.

40. Johnson N.F., Rude T., “Introduction to Steganography Hidden Information”, Regional Computer Forensic Group (RCFG) and Mid-Atlantic Chapter of High-Technology Crime Investigation Association (HTCIA) George Mason University Computer Forensics Symposium (GMU 2001), Fairfax, VA, August 13-17, 2001.

41. Kahn D., “The Codebreakers”, Macmillan, New York, 1967.

42. Katzenbeisser S., Petitcolas F.A.P., “Information Hiding Techniques for Steganography and Digital Watermarking”, Artech House, INC. 685 Canton Street Norwood, MA 02062, 2000.

43. Kessler G.C., “Steganography: Hiding Data within Data”, http://www.garykessler.net/library/steganography.html, September 2001.

44. Kharrazi M., Sencar H.T., Memon N, “Image Steganography: Concepts and Practice”, WSPC/Lecture Notes Series, April 22, 2004.

45. Kim Y.S., Kim Y.M., Choi J.Y., Baik D.K., “Information Hiding System StegoWaveK for Improving Capacity”, International Symposium, ISPA 2003 Aizu-Wakamatsu, Japan, July 2-4, Proceedings, Springer Berlin/ Heidelberg, ISSN 0302-9743, vol. 2745/2003, 2003.

46. Krenn R., “Steganography and steganalysis”, http://www.krenn.nl/univ/cry/steg, 2004.

47. Kutter M., Jordan F., Bossen F., “Digital Signature of Color Images using Amplitude Modulation”, in Proceedings of SPIE storage and retrieval for image and video databases, San Jose, USA, February 13-14, 1997

48. Lai X., “On the Design and Security of Block Ciphers”, ETH Series in Information Processing, v. 1, Konstanz: Hartung-Gorre Verlag, 1992.

49. Lampson B.W., “A Note on the Confinement Problem”, Communications of the ACM, vol. 16. no. 10, pp. 613-615, October 1973.

50. Maroney C.: Hide and Seek v4.1, Freeware. ftp://ftp.csua.berkeley.edu/pub/ cypherpunks/ steganography/ hdsk41b.zip

51. Marvel L.M., Boncelet C.G., Jr. Retter C.T., “Reliable Blind Information Hiding for Images”, in Proceedings of 2nd Workshop on Information Hiding (D. Aucsmith, editor), Lecture Notes in Computer Science, Springer, 1998.

52. Memon N., Wong, P., “Protecting digital media content”, Communications of the ACM, vol 41, no. 7 , pp. 34–43, July 1998.

53. Microsoft Corporation, “Microsoft Windows Programmer's Reference”, vol. 2, v3, Microsoft Press, Redmond, WA, 1990.

54. Moerland T, “Steganography and Steganalysis”, Universiteit Leiden, Rhone-Alpes, 2003.

55. Morkel T., Eloff J.H.P., Olivier M.S., “An Overview of Image Steganography”, in Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005), Sandton, South Africa, June/July 2005 (Published electronically)

56. Murray A.H., Burchfiled R.W (eds.), “The Oxford English Dictionary: Being a Corrected Re-issue”, Oxford, England: Clarendon Pres, 1933.

57. Newman B., “Secrets of German Espionage”, London: Robert Hale Ltd., 1940.

58. Outguess, http://www.outguess.org

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

60. Pfitzmann B., “Information Hiding Terminology”, In Anderson [3], pp. 347-350, ISBN 3-540-61996-8, 1996.

61. Phan R.C.W., Ling H.C., “Steganalysis of Random LSB Insertion Using Discrete Logarithms Proposed At Cıta03”, $M^2$USIC03, PJ, Malaysia, 2-3 October 2003.

62. Pitas I., Nikolaidis N., “Copyright Protection of Images using Robust Digital Signatures”, in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-96), vol. 4, pp. 2168-2171, May 1996.

63. Rivest R.L., “The MD5 Message Digest Algorithm”, Request for Comments (RFC) 1321, Internet Activities Board, Internet PrivacZ Task Force, 3RIPEMD-1281, April 1992.

64. Popa R., “An Analysis of Steganographic Techiques”, Ph.D Thesis, 1998.

65. Potdar V.M., Chang E., “Grey Level Modification Steganography for Secret Communication”, Industrial Informatics, INDIN '04. 2004 2nd IEEE International Conference, pp. 223-228, ISBN: 0-7803-8513-6, 24-26 June 2004.

66. Provos N., “Defending Against Statistical Steganalysis”, 10th USENIX Security Symposium, Washington, DC, 2001.

67. Revelation, http://revelation.atspace.biz/steganography.html.

68. Rijmen V., “Cryptanalysis and Design of Iterated Block Ciphers”, PHd Thesis, October 1997.

69. Sağıroğlu Ş., Tunçkanat M., “A Secure Internet Communication Tool”, Journal of Telecommunication, 2002.

70. Sayood K.: Introduction to Data Compression, Morgan Kauffman Publishers, Inc. 340 Pine Street, Sixth Floor, San Francisco, CA 94104-3205, USA, 1996.

71. Schott G., “Schola steganographica: in classes octo distributa....”, Jobus Hertz, printer, 1680, bound with: Gasparis Schotti...Technica curiosa... Herbipoli, [Courtesy of Whipple Science Museum, Cambridge.], 1665.

72. Secure Engine, http://securengine.isecurelabs.com/

73. Sellars D., “An Introduction to Steganography”, Online book, 1999.

74. Simmons G., “The Prisoners' Problem and the Subliminal Channel", CRYPTO83 Advances in Cryptology, pp. 51-67, Aug 22 -24, 1984.

75. Stego Machine, http://www.fazle.info/downloads.htm.

76. Stella, http://www.stella-steganography.de/overview.html.

77. Stinson D.R., “Cryptography: Theory and Practice, Second Edition”, CRC Press, 2002.

78. Swanson M., Kobayashi M., Tewfik A., “Multimedia Data Embedding and Watermarking Technologies”, Proc. of IEEE, vol. 86, no. 6, pp. 1064-1087, June 1998.

79. Şahin A., Buluş E., Sakallı M.T., “24-Bit Renkli Resimler Üzerinde En Önemsiz Bite Ekleme Yöntemini Kullanarak Bilgi Gizleme”, Trakya Üniversitesi Fen Bilimleri Dergisi, Edirne-Türkiye, 2006.

80. Şahin A., Buluş E., Sakallı M.T., “Gri Seviye Resimler Üzerinde Rasgele LSB Yöntemini ve Sayı Teorisini Kullanarak Bilgi Gizleme ve Steganaliz”, Akademik Bilişim Konferansları 2006-AB2006, Denizli-TÜRKİYE, Şubat-2006.

81. Şahin A, Buluş E., Buluş H.N, Sakallı M.T., “24-bit Renkli Resimler Üzerine Uygulanan RS Steganalizde Maske Seçimlerinin Etkileri” Elektrik Elektronik Bilgisayar Mühendisliği Sempozyumu (ELECO 2006), Bursa-TÜRKİYE, Aralık-2006 (poster).

82. Tacticus A., “How to Survive Under Siege / Aineias the Tactician”, Oxford, England: Clarendon Pres, pp. 84-90 and 183-193, Clarendon Ancient History Series, 1990.

83. Tunçkanat M., Sağıroğlu Ş., “Güvenli İletişim İçin Yeni Bir Yaklaşım: Resim İçerisine Döküman Gizleme”, GAP IV. Mühendislik Kongresi (Uluslararası Katılımlı), Şanlıurfa, vol.1, s.665-668, 6-8 Haziran 2002.

84. Upham D., Jsteg, http://islab.oregonstate.edu/documents/ftpsites/berkeley/jsteg, 2003.

85. Wallace, G. K., “The JPEG Still Picture Compression Standard”, Communications of the ACM, 34(4), 30-44, 1991.

86. Wang H., Wang S., “Cyber Warfare: Steganography vs. Steganalysis”, Communications of the ACM, vol. 47, no. 10, October 2004.

87. Westfeld, A., “High Capacity despite Better Steganalysis (F5–A Steganographic Algorithm)”, In: Moskowitz, I.S. (eds.): Information Hiding. 4th International Workshop. Lecture Notes in Computer Science, Vol.2137. Springer-Verlag, Berlin Heidelberg New York, 289–302, 2001.

88. Westfeld A., Pfitzmann A., “Attacks on Steganographic Systems”, Information Hiding. Third International Workshop, IH'99, Dresden, Germany, September/October, 1999, Proceedings, LNCS 1768, Springer-Verlag Berlin Heidelberg 2000.

89. Zim H.S., “Codes and Secret Writing”, William Morrow, New York, 1948.

# TEZ SIRASINDA YAPILAN ÇALIŞMALAR

## Ulusal Hakemli Dergi Makaleleri

1. A. Şahin, E. Buluş, M.T. Sakallı, "24-Bit Renkli Resimler Üzerinde En Önemsiz Bite Ekleme Yöntemini Kullanarak Bilgi Gizleme", Trakya Üniversitesi Fen Bilimleri Dergisi, Edirne-TÜRKİYE, Haziran–2006.

## Uluslararası Kongre ve Sempozyum Bildirileri

1. A. Şahin, T. Yerlikaya, E. Buluş, E. Akata, "Effect of Cryptography on Cover Object in Steganography", International Scientific Conference UNITECH'06, Gabrovo-Bulgaria, Kasım–2006.
2. M.T. Sakallı, E.Buluş, A.Şahin, F.Büyüksaraçoğlu, "Differential Cryptanalysis for a 3-Round SPN", 4th Internternational Conference on Electrical and Electronics Engineering, ELECO'2005, Bursa-TURKEY, Aralık–2005.

## Ulusal Kongre ve Sempozyum Bildirileri

1. A. Şahin, E. Buluş, H.N. Buluş, M.T. Sakallı, "24-bit Renkli Resimler Üzerine Uygulanan RS Steganalizde Maske Seçimlerinin Etkileri" Elektrik Elektronik Bilgisayar Mühendisliği Sempozyumu (ELECO 2006), Bursa-TÜRKİYE, Aralık–2006 (poster).
2. M.T. Sakallı, E. Buluş, A. Şahin, F. Büyüksaraçoğlu, "S kutularında doğrusal eşitlik-Affine Equivalence in S-boxes", IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı-SİU 2006, Antalya-TÜRKİYE, Nisan–2006.

3. M.T. Sakallı, E. Buluş, A. Şahin, F. Büyüksaraçoğlu, H.N. Buluş, "AES S Kutusuna Benzer S Kutuları Üreten Simulatör", Akademik Bilişim Konferansları 2006-AB2006, Denizli-TÜRKİYE, Şubat–2006.
4. A. Şahin, E. Buluş, M.T. Sakallı, "Gri Seviye Resimler Üzerinde Rasgele LSB Yöntemini ve Sayı Teorisini Kullanarak Bilgi Gizleme ve Steganaliz", Akademik Bilişim Konferansları 2006-AB2006, Denizli-TÜRKİYE, Şubat–2006.
5. A. Şahin, E. Buluş, M.T. Sakallı, "Modern Blok Şifreleme Algoritmalarının Gücünün İncelenmesi", II. Mühendislik Bilimleri Genç Araştırmacılar Kongresi-MBGAK'2005, İstanbul-TÜRKİYE, Kasım–2005.
6. M.T. Sakallı, E. Buluş, A. Şahin, F. Büyüksaraçoğlu, "AES S Kutusuna Benzer 4-Bit Girişe Ve 4-Bit Çıkışa Sahip S Kutularının Tasarımı", II. Mühendislik Bilimleri Genç Araştırmacılar Kongresi-MBGAK'2005, İstanbul-TÜRKİYE, Kasım–2005.
7. M.T. Sakallı, E. Buluş, A. Şahin, F. Büyüksaraçoğlu, "Bir Blok Şifreleme Algoritmasına Karşı Square Saldırısı", Ağ ve Bilgi Güvenliği Ulusal Sempozyumu-ABG2005, İstanbul-TÜRKİYE, Haziran–2005.

# ÖZGEÇMİŞ

Andaç ŞAHİN, 2 Temmuz 1978 yılında Kırklareli'nde doğdu. İlk, Orta ve Lise öğrenimini Malkara/Tekirdağ'da tamamladıktan sonra 1994 yılında Trakya Üniversitesi Mühendislik Mimarlık Fakültesi Bilgisayar Mühendisliği bölümünü kazandı ve bu bölümden 1998 yılında mezun oldu. 1998 yılı Ekim ayında Trakya Üniversitesi Mühendislik Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışmaya başlayan Andaç ŞAHİN aynı sene Yüksek Lisans çalışmalarına başladı. Yüksek lisansını 2001 yılında başarıyla bitirdi. 2002 yılında Trakya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dalı'nda doktora çalışmalarına başladı. Andaç ŞAHİN halen Trakya Üniversitesi Bilgisayar Mühendisliği Bölümü'nde Araştırma Görevlisi olarak görev yapmaktadır.