T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KAYIPSIZ VERİ SIKIŞTIRMA VE STEGANOGRAFİ TEKNİĞİNE DAYALI YENİ BİR YÖNTEM

YÜKSEK LİSANS TEZİ

Ertuğrul DUMAN

Enstitü Anabilim Dalı	**:**	**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Prof. Dr. Cemil ÖZ**

Ağustos 2019

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KAYIPSIZ VERİ SIKIŞTIRMA VE STEGANOGRAFİ TEKNİĞİNE DAYALI YENİ BİR YÖNTEM

YÜKSEK LİSANS TEZİ

Ertuğrul DUMAN

Enstitü Anabilim Dalı	:	**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**
Enstitü Bilim Dalı	:	**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**

Bu tez 05.08.2019 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.

Prof. Dr.	**Prof. Dr.**	**Doç. Dr.**
Şeref SAĞIROĞLU	**Cemil ÖZ**	**Numan ÇELEBİ**
Jüri Başkanı	**Üye**	**Üye**

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Ertuğrul DUMAN

05.08.2019

TEŞEKKÜR

Yüksek lisans tez çalışması boyunca bilgi ve deneyimlerinden yararlandığım değerli danışmanım Prof. Dr. Cemil ÖZ'e, yine bu çalışmanın ortaya çıkmasında fikir ve önerilerini esirgemeyen Prof. Dr. Şeref SAĞIROĞLU'na, çalışmalarım süresince her türlü desteği sağlayan eşim Öğr. Gör. Burcu DUMAN'a, varlıkları ile bana güç, neşe ve enerji veren çocuklarım Emirhan ve Ömer Kerem'e, teşşekür ve şükranlarımı sunarım.

İÇİNDEKİLER

TEŞEKKÜR i
İÇİNDEKİLER ii
SİMGELER VE KISALTMALAR LİSTESİ v
ŞEKİLLER LİSTESİ vii
TABLOLAR LİSTESİ ix
ÖZET x
SUMMARY xi

BÖLÜM 1.
GİRİŞ 1
1.1. Tez Organizasyonu 3

BÖLÜM 2.
KRİPTOLOJİ 4
2.1. Kriptoloji Temelleri 5
2.2. Kriptolojide Şifreleme Yöntemleri 7
2.2.1. Yer değiştirme (Transposition) yöntemi 8
2.2.2. Yerine koyma (Substitution) yöntemi 8
2.2.3. Dizi/Katar (Akış) şifeleme yöntemi 10
2.2.4. Blok şifreleme yöntemi 10
2.3. Şifreleme Algoritmaları ve Sınıflandırılması 14
2.3.1. Karma şifreleme algoritmaları 16
2.3.2. Simetrik ve asimetrik şifreleme algoritmalarının karşılaştırılması 17
2.3.3. Kriptografik yapıların kullanım ve uygulama alanları 18
2.3.4. Kriptoanaliz 18

2.3.4.1. Kripto sistemlere karşı yapılan saldırı türleri 19
BÖLÜM 3.
STEGANOGRAFİ 21
3.1. Steganografi Tarihçesi 21
3.2. Steganografi Temelleri ve Gereksinimleri 24
3.3. Kriptografi ve Steganografi Arasındaki Farklar 26
3.4. Filigranlama/Damgalama ve Steganografi Arasındaki Farklar 27
3.5. Steganografide Kullanılan Temel Teknikler 28
3.5.1. Uzamsal alan teknikleri 29
3.5.2. Dönüşüm alan teknikleri 29
3.5.3. Sıkıştırılmış alan teknikleri 29
3.6. Steganografi Sınıflandırılması 30
3.6.1. Metin (Text) steganografi 31
3.6.2. Resim (İmage) steganografi 35
3.6.3. Ses (Audio) steganografi ve kullanılan teknikler 39
3.6.3.1. Düşük bit kodlaması (Least significant bit-LSB) 43
3.6.3.2. Aşama/Faz kodlaması (Phase coding) 44
3.6.3.3. Tayf yayılması (Spread spectrum) 45
3.6.3.4. Yankı veri saklaması (Echo data hiding) 47
3.7. Steganografik Yazılımlar 48
3.8. Steganaliz (Sır açma) 49
3.8.1. Steganalizde saldırı türleri 50
3.8.2. Steganalizde kullanılan ölçüm yöntemleri 51
3.8.3. Steganaliz teknikleri ve mevcut steganaliz yazılımları 54
3.8.3.1. $\chi 2$ (ki-kare) testi 55
BÖLÜM 4.
MATERYAL VE GELİŞTİRİLEN UYGULAMA 58
4.1. Ses Dalgası 59
4.1.1. Ses dalgasının yapısı ve özellikleri 59
4.1.2. Ses sinyalinin sayısal formata dönüştürülmesi 61
4.2. WAV Dosya Yapısı 65

4.3. ASCII Karakter Tablosu 69
4.4. Veri Sıkıştırma Teknikleri ve LZW Algoritması 72
4.5. Geliştirilen Uygulama 74
4.5.1. Veri kodlama ve ses dosyası içerisine gizleme 75
4.5.2. Ses dosyasından veri çıkarma ve kod çözme 85
4.6. EDStego yazılımı ve diğer stego yazılımların karşılaştırlması 90

BÖLÜM 5. U
YGULAMA ANALİZLERİ VE SONUÇ 93
5.1. Kapasite Analizi 93
5.2. Algılanamazlık Analizleri 95
5.2.1. $\chi 2$ (ki-kare) analizi 98
5.2.2. Diğer analizler 105
5.3. Değerlendirme ve Sonuç 111

KAYNAKLAR 114
ÖZGEÇMİŞ 120

SİMGELER VE KISALTMALAR LİSTESİ

3DES	:Triple Data Encryption Standard (Üçlü Veri Sifreleme Standardı)
AES	:Advanced Encryption Standard (Gelismis Sifreleme Standardı)
ASCII	:American Standard Code Information Interchange (Amerikan Kod Bilgi Takası Standardı)
AVI	:Ham Görüntü Standardı(Audio Video Interleave)
BMP	:Ham Resim Standardı(Bitmap)
DB	:Desibel (Gürültü, Sinyal, Voltaj, Elektromanyetik Dalga vb.) Ölçü Birimi
DCT	:Discrete Cosine Transform (Ayrık Kosinüs Dönüşümü)
DES	:Data Encryption Standard (Veri Sifreleme Standardı)
DFT	:Discrete Fourier Transform (Ayrık Fourier Dönüşümü)
DIS	:Data Into Sound (Ses Dçine Veri)
DSA	:Digital Signature Algorithm (Dijital İmza Algoritması)
EXE	:Windows Çalıştırılabilir Dosya Standardı (Executable)
GIF	:Graphics Interchange Format (Sıkıştırılmış Resim Standardı)
HSL	:Renk tonu, Doyum, Açıklık(Hue, Saturation, Lightness)
HSV	:Renk Tonu, Doyum, Değer (Hue, Saturation, Value)
HTML	:Hyper Text Markup Language (Üstün Metin Dsaretleme Dili)
HZ	:Hertz (Heinrich Rudolf Hertz)
JPEG	:Joint Photographic Experts Group (Birlesik Fotoğrafik Uzmanlar Grubu)
JPG	:Joint Photographic Group (Sıkıştırılmış Resim Standardı()
KB	:Kilo Byte (Kilo Bayt)
KHZ	:Kilo Hertz

LSB	:Least Significant Bit (En az Önemli Bit)
MAE	:Mean Absolute Error (Ortalama Mutlak Hata)
MD5	:Message Digest 5
MIDI	:Musical Instrument Digital Interface (Müzikal Enstrüman Sayısal Ara yüzü)
MP3	:MPEG-1 Audio Layer-3 (Sıkıştırılmış Ses Standardı)
MPEG	:Moving Picture Experts Group (Video Sıkıştırma Standardı)
MSE	:Mean Squared Error (Ortalama Karesel Hata)
NIST	:National Institute of Standards and Tehcnology (Ulusal Standartlar ve Teknoloji Enstitüsü)
NSA	:National Security Agency (Ulusal Güvenlik Ajansı)
PCM	:Pulse Code Modulation (Titresim Kod Modülasyonu)
PDF	:Portable Document Format (Tasınabilir Doküman Biçimi)
PNG	:Portable Network Graphic (Tasınabilir Ağ Grafiği)
PSNR	:Peak Signal-to Noise Ratio (Doruk Sinyal Gürültü Oranı)
RGB	:Red Green Blue (Kırmızı Yesil Mavi)
RIFF	:Resource Interface File Format (Ham Ses Standardı)
RSA	:Ron Rives, Adi Shamir, Leonard Aldeman
SHA	:Secure Hash Algorithm (Güvenli Özetleme Algoritması)
SNR	:Signal-to Noise Ratio (Sinyal Gürültü Oranı)
TXT	:Metin Dosya Standardı
WAV	:WAVeform ses biçimi (Ham Ses Standartı)
XML	:EXtensible Markup Language (Genişletilebilir İşaretleme Dili)
XOR	:eXclusive OR (Harici Mantıksal Veya İşlemi)

ŞEKİLLER LİSTESİ

Şekil 2.1. Kriptoloji temel bileşenleri [5] 5
Şekil 2.2. Dizi/Katar (Akış) şifreleme yöntemi [5] 10
Şekil 2.3. Blok şifreleme yöntemi [5] 11
Şekil 2.4. Feistel şifreleme tekniği [17] 12
Şekil 2.5. Yer değiştirme permutasyon (SPN) ağları çalışma tekniği [17] 13
Şekil 2.6. Şifreleme algoritmalarının yöntem ve anahtar kullanımına göre sınıflandırılması 15
Şekil 3.1. Stegosistem yapısı, temel bileşenleri ve gereksinimleri 25
Şekil 3.2.Steganografinin uygulama ortamları, kullanılan yöntem ve tekniklere göre sınıflandırılması 31
Şekil 3.3. DUMAN Verisinin HTML etiketleri içerisine gizlenmesi 34
Şekil 3.4. RGB, CMYK, HSV, YUV Renk uzayları [36] 36
Şekil 3.5. Ses iletim yöntemleri [45] 40
Şekil 3.6.Ses dosyalarında LSB tekniği [49] 43
Şekil 3.7. Aşama/Faz kodlaması [50] 45
Şekil 3.8. Tayf yayılması genel işleyiş adımları [50] 46
Şekil 3.9. Yankı veri kodlaması [45] 47
Şekil 3.10. Resim içerisine veri gizlenmeden önce ve sonraki frekans dağılımları 56
Şekil 4.1. Ses dalgası ve parametreleri 60
Şekil 4.2. Ses sinyalindeki tepe ve çukur noktaların yoğunluğu 61
Şekil 4.3. Analog sinyalin sayısal formata dönüştürülme adımları [58] 62
Şekil 4.4. ADC ve DAC dönüştürme 63
Şekil 4.5. Ses sinyalinin PCM ve DSD tekniklerine göre sayısal formata dönüştürülmesi [71] 65
Şekil 4.6. WAV dosya yapısı [72] 66

Şekil 4.7. Örnek WAV dosyası[72] 68
Şekil 4.8. Standart ASCII tablosu [65] 70
Şekil 4.9. Genişletilmiş ASCII tablosu [65] 71
Şekil 4.10. Veri sıkıştırma, şifreleme, gizleme işlem adımları 76
Şekil 4.11. Uygulama veri sıkıştırma, şifreleme ve gizleme arayüzü 77
Şekil 4.12. Sözlük oluşturma ve veri sıkıştırma C# kod parçası 78
Şekil 4.13. Orjinal mesaj boyutları ve sıkıştırma sonrası elde edilen yeni boyutlar 79
Şekil 4.14. Kodlanmış mesaj verisinin üzerine kontrol bitlerinin eklenmesi 81
Şekil 4.15. Uygulama veri çıkarma ve kod çözme ara yüzü 86
Şekil 4.16. Veri çıkartma, şifre çözme, veri açma işlem adımları 87
Şekil 4.17. Stego dosyadan veri çıkaran uygulama kod parçası 88
Şekil 4.18. Tersinden sözlük oluşturan ve gizli mesajı stego dosyadan çıkaran kod parçası 89
Şekil 5.1. SNR, PSNR Değerleri hesaplama fonksiyonu matlab kodu 98
Şekil 5.2. $\chi 2$ (ki-kare) testi matlab uygulama kodları 102
Şekil 5.3. Farklı veri gömme tekniği kullanılarak elde edilen stego nesneye ait ki-kare analiz sonuçları 103
Şekil 5.4. Orjinal ses dosyasına ait ki-kare analizleri ve değer çiftlerinin gösterimi 103
Şekil 5.5. Stego nesnelere ait ki-kare analiz sonuçları 104
Şekil 5.6. Diğer analizlere (Histogram, Genlik, Spektrogram, DÇ, Otokorelasyon) ait analizleri matlab kod parçası 107
Şekil 5.7. Orjinal nesneye ait diğer (Histogram, Genlik, Spektrogram, DÇ, Otokorelasyon) analiz sonuçları 108
Şekil 5.8. Stego6 nesneye ait diğer (Histogram, Genlik, Spektrogram, DÇ, Otokorelasyon) analiz sonuçları 109
Şekil 5.9. Stego8 nesneye ait diğer (Histogram, Genlik, Spektrogram, DÇ, Otokorelasyon) analiz sonuçları 110

TABLOLAR LİSTESİ

Tablo 2.1. Anahtar uzunluğuna göre şifre çözme süreleri [10] 6
Tablo 2.2. Türkçe ve İngilizce harflerin bağıl frekans dağılımları [5]..... 9
Tablo 2.3. Simetrik ve asimetrik şifreleme algoritmalarının genel özellikleri 17
Tablo 2.4. Simetrik ve asimetrik şifreleme algoritmalarının özelliklerinin karşılaştırılması [10]..... 17
Tablo 2.5. Kriptografik sistemler ve uygulama alanları 18
Tablo 2.6. Şifreli mesajlarda saldırı türleri [5]..... 19
Tablo 2.7. Kaba kuvvet saldırılarında şifre kırmak için gereken ortalama süreler [5]..... 20
Tablo 3.1. Steganografi ve kriptografi arasındaki farklar. 27
Tablo 3.2. DUMAN Versinin kodlanması 33
Tablo 3.3. Ses steganografisinde kullanılan yöntemlerin karşılaştırılması [49] 42
Tablo 3.4. Steganografik yazılımlar [64] 48
Tablo 3.5. Steganaliz saldırı çeşitleri [5, 46, 55] 51
Tablo 4.1. WAV dosya yapısı yığın açıklamaları [72] 67
Tablo 4.2. Yazdırılmayan ASCII kodlarının anlamları..... 71
Tablo 4.3. LZW algoritması kodlama örneği..... 74
Tablo 4.4. Örnek bir ses dosyasının ilk 44 byte'lık verileri ve değerleri..... 82
Tablo 4.5. Örnek bir ses dosyasının ilk 72 byte'lık verileri..... 82
Tablo 4.6. EDStego yazılımı ile diğer stego yazılımlarının karşılaştırılması 91
Tablo 5.1. Veri sıkıştırma kapasite analiz sonuçları 94
Tablo 5.2. Uygulama işlem süreleri, taşıyıcı ve stego dosya boyutları..... 94
Tablo 5.3. Uygulama SNR, PSNR performans değerleri 97

ÖZET

Anahtar Kelimeler: Kriptografi, Kriptoanaliz, Veri Sıkıştırma, LZW, LSB, Steganografi, Ses Steganografi, Steganaliz, SNR, PSNR, Ki-Kare

Bu tezde sunulan çalışmanın temel hedefi; steganografi'nin (bilgi gizleme sanatı) temel ilkelerine (algılanamazlık, sağlamlık, kapasite) bağlı kalarak, etkin bir şifreleme (krpito) algoritması uygulayıp, güvenli bir sayısal iletişim sağlamaktır. Bu bağlamda gizlilik gerektiren metnin ya da dosyanın boyutunu kayıpsız bir şekilde küçültüp (sıkıştırıp), güçlü bir şekilde şifreleyerek ses dosyası içerisine gömülmesini ve tersi adımları uygulayarak sıkıştırılmış ve şifrelenmiş gizli bilginin ses dosyasından çıkarılması sağlanmıştır. İnsan duyu organlarından işitme duyusunun görme duyusuna göre daha hassas olduğu gözönüne alınırsa, ses dosyaları üzerinde yapılacak olan veri gizleme çalışmalarının oldukça zor bir çalışma alanı olduğu söylenilebilir. Ses steganografisi alanında Dünyada ve Türkiyede yapılan bilimsel çalışmalar ve akademik yayınlar diğer alanlardaki steganografi çalışmalarına göre daha az sayıda olduğu görülmektedir.

Yapılan bu çalışmada, sıkıştırma işlemi ile taşıyıcı ortamın (video, ses, resim, metin) saklama kapasitesi sorunu kısmen aşılırken, simetrik şifreleme algoritma kullanılmasıyla da muhtemel kriptoanaliz ve steganaliz (sır açma) saldırılarına karşı korunabilirliği (sağlamlığı) artırılmıştır. Bu amaçla gizlenmek istenen bilgi kayıpsız veri sıkıştırma algoritmalarından olan sözlük tabanlı kodlama yöntemlerinden LZW algoritmasının genişletilmiş ve geliştirilmiş yeni haliyle kodlanıp sıkıştırılmıştır. Veri güvenliği ve mahremiyetini sağlamak için simetrik şifreleme algoritmaları uygulama içerisinde bir seçenek olarak kullanıcıya bırakılmıştır. Elde edilen yeni bilgi, stego anahtar kullanarak taşıyıcı nesnenin boyutunda herhangi bir değişim olmaksızın en az önemli bitlerine (Least Significant Bit - LSB) rastgele yerleştirilmiştir. Bu işlem sonucunda insan duyu organlarının algısından korunması (algılamazlık) sağlanmıştır. Tersi adımları yaparak alıcı tarafta gizli bilgi kayıpsız olarak geri elde edilmiştir.

Kriptografi, steganografi ve veri sıkıştırma teknikleri ile elde edilen yeni yapı hakkındaki veri gizliliği başarısını değerlendirmek için, SNR, PSNR ölçüm değerleri verilmiş ve başta $\chi 2$ (ki-kare) testi olmak üzere çeşitli sır açma (steganaliz) tekniklerinden elde edilen sonuçlar da sunulmuştur.

A NOVEL TECHNIQUE BASED ON LOSSLESS DATA COMPRESSION AND STEGANOGRAPHY

SUMMARY

Keywords: Cryptography, Cryptoanalysis, Data Compression, LZW, LSB, Steganography, Audio Steganography, Steganalysis, SNR, PSNR, Chi-Sequare

The main objective of the work presented in this thesis is; to implement an efficient crypto (crypto) algorithm to provide a secure digital communication by adhering to the basic principles of steganography (knowledge concealment art) (imprecision, robustness, capacity). In this context, to compress (compress) the size of the text or file which is required for confidentiality, to be embedded in the audio file with strong encryption, and vice versa, to extract the compressed and encrypted secret information from the audio file. When it is considered that the sense of hearing from the human senses is more sensitive than the sense of sight, it can be said that the data concealment work on audio files is a very difficult work area. It is seen that, scientific research and academic publications in the field of audio steganography, are fewer than other studies in the field of steganography.

In this study, the protection (robustness) against the possible cryptoanalysis and steganalysis attacks is increased by an efficient and layered encryption algorithm, while the storage capacity problem of the media (video, audio, picture, text) is partially overcome by the compression process. For this purpose, the information to be concealed is encoded and compressed with LZW algorithm of dictionary based coding methods, which is one of the lossless data compression algorithms. Symmetric encryption algorithms are left to the user as an option within the application to ensure data security and privacy. The new information obtained is randomly placed in the least significant bits (LSB) without any change in the size of the carrier object using the stego key. As a result of this process, it is protected from the perception of other human sense organs. By doing the reverse steps, the secret information on the receiver side is recovered as lossless.

SNR, PSNR measurement values are given to evaluate the success of data confidentiality about the new structure obtained by cryptography, steganography and data compression techniques and the results obtained from various tests ($\chi 2$ (chi-sequare), spectrogram, spectral, histigram) of steganalysis are also presented.

BÖLÜM 1. GİRİŞ

Bilgi; evrenin varoluşundan bu güne kadar insanoğlunun sahip olduğu en değerli varlık olarak karşımıza çıkmaktadır. Bireyler ve kurumlar kendi aralarında bilgi akışını ve iletişimi sağlayabilmek için geçmişten günümüze birçok farklı yöntem geliştirmiş ve kullanmışlardır (tamtam-davul, duvar resimleri, duman, mektup, posta güvercinleri, ulak-haberci, telgraf, telefon, bilgisayar, internet v.b.) [1, 2, 3]. Bütün bu haberleşme ve iletişim yöntemleri içinde gizlilik ve güvenlik ayrı bir önem oluşturmuştur [4].

Teknolojinin hızla geliştiği ve sürekli yenilendiği günümüz dünyasında, iletişim yöntem ve şekilleri de değişerek tamamen elektronik sayısal (dijital) ortamlara taşınmıştır. Günlük yaşantımızın bir parçası olan akıllı telefonlar, tabletler, bilgisayarlar, bankalar, bankamatikler, askeri sistemler, yeni nesil oyuncaklar, işletmelerde kullanılan tezgâhlar, robotlar, akıllı evler için kullanılan elektronik aygıtlar internet veya intranet üzerinden sürekli birbirleri ile etkileşim ve iletişim halindedir. Bu sayede veri iletişimi/paylaşımı hızlı ve etkin şekilde yapılabilmektedir. Bu gelişmeler insanların bulundukları yerden bağımsız olarak istedikleri bilgiye hızlı bir şekilde ulaşmasına imkân sağlarken, aynı zamanda eldeki bilginin istenmeyen üçüncü kişiler tarafından da erişimini kolaylaştırmaktadır.

Bilginin bu denli önem kazandığı, iletişim yöntemlerinin sayısal ortamlarda yapıldığı günümüzde ve sonrasında ortaya çıkan temel ihtiyaçların başında veri iletişiminin ve haberleşmenin üçüncü şahısların algı ve tehdidinden uzak güvenli bir şekilde yapılabilmesini sağlamaktır. Ortak ağlar kullanılarak gerçekleştirilen haberleşme işlemlerinde ortaya çıkan en önemli sorunların başında bu haberleşmenin üçüncü taraflar tarafından izlenebilir ve sabote edilebilir olması gerçeği gelmektedir. Saldırılardan ve tehditlerden korunma araçları içerisinde en sık kullanılan yöntem şifreleme tekniğidir. [1, 4]. Şifreleme teknolojisi iletilmek istenilen veriyi belirli bir

yapıda değiştirerek/bozarak istenmeyen üçüncü şahısların anlamayacağı şekle dönüştürürler. Ancak bu şifreleme sonrasında yapısı değiştirilmiş olan veri/veriler üçüncü kişiler tarafından fark edilmesi sonucu ilgili veri üzerinde farklı analiz ve yöntemlerle (kriptoanaliz) gizli bilgi deşifre edilebilmektedir [4]. Ayrıca şifreli haberleşme yapıldığı tespit edilmesi durumunda iletişim kanalı sabote edilip kesintiye uğratılabilir. Bu durumda iletilmek istenilen bilginin karşı-alıcı tarafa sağlıklı bir şekilde aktarılabilmesi için tek başına şifreleme yönteminin yetersiz kalacağı anlaşılmaktadır.

Daha sağlıklı ve güvenli bir iletişim sağlayabilmek için iletilmek istenilen gizli bilginin fark edilmeden alıcı tarafa aktarılması ihtiyacı ortaya çıkmaktadır. Bunu sağlamak içinde veri gizleme/sır örtme (steganografi) yöntemleri geliştirilmiş ve kullanılmıştır [4]. Steganografi işlemi ile iletilmek istenilen veriyi başka bir veri içerisine (taşıyıcı ortam: video, ses, resim, metin) saklayarak/gömerek alıcı tarafa aktarılmasını sağlanmaktadır. Bu sayede aktarılmak istenilen bilgi insan duyu organlarının algısından uzaklaştırılarak karşı tarafa iletilebilir. Ancak bu yöntem de yapılabilecek muhtemel veri çıkarma/sır açma (steganaliz) ataklarına karşı tek başına güvenli bir iletişim sağlamayacaktır. Ayrıca taşıyıcı ortam olarak kullanılan öğelerin, kullanılan steganografi tekniğine bağlı olarak değişen, belirli bir veri saklama kapasitesi olacaktır. Bu durumda başvurulan diğer bir yöntem gönderilecek mesajın boyutunun sıkıştırılmsı veya küçültülmesi olarak karşımıza çıkmaktadır.

Mevcut şifreleme ve veri gizleme yöntemlerinin tek başına kullanılmaları istenilen korumayı ve güvenliği sağlayamadıkları durumlar ortaya çıkacaktır. Bu durum da veri iletişiminde katmanlı güvenlik ve gizlilik adımlarının kullanılmasını kaçınılmaz hale getirmiştir. Yapılan bu çalışma da yukarıda belirtilen eksiklik veya korunma açıklarını en aza indirebilmek için katmanlı güvenlik adımlarını barındıran veri sıkıştırma, veri şifreleme ve veri gizleme yöntemleri birlikte kullanılarak bir uygulama arayüzü geliştirilmiştir. Geliştirilen uygulama sayesinde veri sıkıştırma, veri şifreleme ve veri gizleme işlemleri tek başlarına (diğer işlemlerden bağımsız) olarak yapılabildiği gibi, bu işlemler birlikte kullanılarak kombine bir yapı elde

edilebilmektedir. Geliştirilen uygulamaya kaynak kodları ile beraber; https://websitem.gazi.edu.tr/site/dumanertugrul/files/EDstego adresinden ulaşılabilir.

1.1. Tez Organizasyonu

Etkin ve sağlıklı bir veri şifreleme ve gizleme yöntemini geliştirmeyi amaçlayan bu çalışma beş temel bölümden oluşmaktadır. İlk bölümde yapılan çalışmanın gereksinimi anlatılmıştır. İkinci bölümde kriptoloji biliminin temelleri ve şifreleme algoritmaları, bu algoritmlarda kullanılan şifreleme teknikleri, kriptografik yapıların uygulama alanları ve kriptagrafik sistemlere karşı yapılan saldırı (kriptoanaliz) yöntemleri anlatılmıştır. Üçüncü bölümde veri gizleme/sır örtme (steganografi) temelleri ve yöntemleri ile beraber veri çıkarma/sır açma (steganaliz) tekniklerine değinilmiştir. Ayrıca steganografik yapılar üzerinde kullanılan ölçüm ve değerlendirme kriterlerine de üçüncü bölüm içerisinde yer verilmiştir. Geliştirilen uygulama ve bu uygulama içerisinde kullanılan ham ses (wav) dosya yapısı, LZW sıkıştırma algoritması dördüncü bölümde ele alınmıştır. Beşinci bölümde ise geliştirilen steganografi sisteminin analizleri yapılarak başarısı değerlendirilmiştir. Yine bu bölümde veri sıkıştırma ve veri gizleme üzerine yapılan bu çalışma bağlamında tartışma ve sonuç bilgilerine yer verilmiştir.

BÖLÜM 2. KRİPTOLOJİ

Kriptoloji, temeli matematik bilimi olan ve genelde sayılar teorisi üzerine kurulmuş şifre bilimidir. Özet olarak kriptoloji, farklı gönderilerin, mesajların belli bir algoritmaya göre şifrelenmesi, bu iletilerin tehdit ve algıdan uzak bir kanal ile alıcıya gönderilmesi ve gönderilen şifreli bilgilerin geri çözülme sürecini bünyesinde barındırır [4]. Veri alışverişi yapan tarafların bu haberleşmeyi emniyetli bir şekilde gerçekleştirmesini amaçlar, bu amaç içinde karmaşık matematiksel yöntemler ve algoritmalar kullanır. Kriptoloji bilimi kendi içerisinde kriptografi ve kriptoanaliz olmak üzere iki alt bölümden oluşmaktadır. Bu bölümler;

- Kriptoanaliz + Kriptografi = Kriptoloji

Kriptografi; bilginin görünümünü, şeklini veya yapısını değiştiren teknik demektir. Yunan dilindeki “kriptos” ve “graphi” kelimerinden türetilmiştir. Kriptografi bilgi güvenliği ile uğraşır. Bir açık metinin bir şifreleme algoritması yardımıyla, karmaşık matamatiksel uygulamalar kullanarak anlaşılamaz hale getirilmesi olarak ta tanımlanabilir [4].

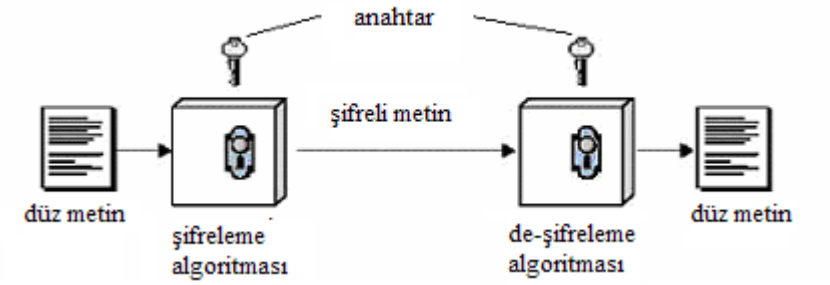
Kriptoanaliz; kriptografik yapıları çözümlemek veya bu yapıları analiz etmek olarak tanımlanabilir. Bir başka ifadeyle kriptografi teknikleri ile elde edilen şifreli sistemi analiz ederek, bu sistemin/yapının güçlü ve zayıf taraflarını belirlemeyi amaçlayan bilim dalıdır. Kriptoanalistler genelde şifre çözmeye dayalı çalışırlar. Bununla birlikte Kriptoanaliz, şifreli bir sistemin çözümlenmeye karşı dayanıklılığının testi şeklinde de yorumlanabilir. Burada dayanıklılık/sağlamlık, şifreli yapının çözülememesi olarak karşımıza çıkmaktadır. Şifreli yapının çözülememesi ise şifrelemede kullanılan gizli anahtarın açığa çıkartılamaması anlamına gelmektedir. Şifrelemede kullanılan gizli anahtarın değişik ataklar karşısında hangi sürede, ne

kadar kaynak harcayarak ve ne kadar veri ile elde edilebilir olduğunu bulmakla ilgilidir.

2.1. Kriptoloji Temelleri

Şifreleme işlemi bir metnin ilgili alıcı dışında başkası tarafından anlaşılamaz hale dönüştürme adımlarını kapsamaktadır. Kullanılan algoritmanın çift yönlü çalışmasına bağlı olarak alıcı tarafta şifreleme adımlarının tersi uygulanarak özgün metin geri elde edilmektedir. Sadece tek yönlü çalışan şifreleme algoritmalarıda (özet algortimalar) vardır. Bu algoritmalar gizli bilgi iletmek için değil genellikle kimlik doğrulama işlemleri için kullanılan algoritmalardır. İlerleyen bölümlerde ayrıca anlatılmıştır. Şifreleme ve şifre çözme genelde bir anahtar (key) kullanılarak yapılır [5]. Kriptografi temel bileşenleri Şekil 2.1.'de gösterilmiştir

Anhtar; kriptografik yapıların veri şifreleme ve tersi şifre çözme adımlarında kullanılan farklı uzunluklarda olabilen sayı dizisidir. Şifrelemede kullanılan anahtarın uzunluğuna bağlı olarak kriptografik yapının güvenliğ/gizliliği artmaktadır.



Şekil 2.1. Kriptoloji temel bileşenleri [5]

Kriptografik bir yapı içerisinde şifrelenecek bilgi düz-metin (plaintext) şeklinde ifade edilirken, Şifreleme işlemi (encryption); mesajın karşı/alıcı taraf dışında hiç kimsenin çözemeyeceği/okuyamayacağı bir şekilde kodlamak olarak nitelendirilmektedir. Yine bu yapı içerisinde kodlanmış mesaja şifreli-mesaj (ciphertext) denmektedir. Şifre çözme (decryption) adımı ise kodlanmış mesajı çözümleyip orijinal haline dönüştürme işlemlerini kapsar. Orijinal veriyi

kodlama/şifreleme ve tersi adımlarını işleterek şifreli mesajı çözümleme adımlarının tamamına ise şifreleme algoritması denilmektedir [5].

Tablo 2.1. Anahtar uzunluğuna göre şifre çözme süreleri [10]

Key Lengt (n bit)	**Oluşabilecek Değer Adedi (2^n)**	**Ort. Çözme süresi 10^6 şifre/s hızında**	**Ort. Çözme süresi 10^9 şifre/s hızında**	**Ort. Çözme süresi 10^{12} şifre/s hızında**
32	~$4x10^9$	36 dk.	2.16 sn	2.16 m sn
40	~10^{12}	6 gün	9 dk	1 sn
56	~$7.2x10^{16}$	1142 yıl	1 yıl 2 ay	10 sa
64	$1.8x10^{19}$	292 000 yıl	292 yıl	3.5 ay
128	$1.7x10^{38}$	$5.4x10^{24}$ yıl	$5.4x10^{21}$ yıl	$5.4x10^{18}$ yıl

Bir algoritmanın kriptografik güvenliği için kullanılan anahtarın uzunluğu ve şifrelenme yöntemi büyük önem taşır. Kullanılan anahtarın bit olarak uzunluğu için n tanımlaması yaparsak; sağlanacak olan güvenlik n sayısının bit uzunluğuyla doğru orantılıdır. Güvenlik açığı oluşturması veya istenilen korumayı sağlayamaması sebebiyle standart olarak kabul edilen 512 bit anahtar uzunluğu yerini 2048 bit uzunluğundaki anahtarlara bırakacağı öngörülmektedir. Tablo 2.2.'de farklı uzunluklardaki anahtarlar için farklı şifre çözme hızına (saniyede bir milyon, bir milyar, bir trilyon) sahip bilgisyaralara göre anahtar bulma süreleri gösterilmiştir.

Etkin bir şifreleme algoritmasının bünyesinde barındırması gereken beş ana fonksiyonu vardır. Bu fonksiyonlar [11]:

- Gizlilik (Confidentiality): Başkaları özgün mesajı görememeli, bilgi sadece istenen kişiler tarafından anlaşılmalıdır.
- Kimlik tanılama/doğrulama (Authentication): Gönderen ve alıcı, birbirlerinin kimliklerini doğrulayabilmesi gerekir.
- Veri Bütünlüğü (Integrity): Verinin gizlenmesi veya iletimi esnasında, fark edilmeden mesaj üçüncü kişiler tarafından değiştirilemez. Veri ilgili/yetkili taraflar dışında kimse tarafından değiştirilmemelidir.
- İnkâr edememe - Reddedilememezlik (Non-repudiation): Mesajın kim tarafından oluşturulduğu veya gönderildiği belirli olmalı ve inkar edilememeli.

- Erişilebilirlik: İlgili tarafların (gönderen-alan) istenilen zamanda istenilen veriye ulaşabilmesi ilkesidir.

2.2. Kriptolojide Şifreleme Yöntemleri

Geçmişten günümüze veri şifreleme ve güvenli bilgi akışı için çeşitli yöntemler geliştirilmiş ve geliştirilen bu yöntemler kriptografik algoritmalar içerisinde kullanılmıştır. Bir sonraki alt bölümde şifreleme algoritmaları kullandıkları bu yöntemlere göre ayrıca sınıflandırılmışlardır. Uygulanış yöntem ve tekniklerine göre şifreleme algoritmaları; klasik şifreleme ve modern şifreleme algoritmaları olarak karşımıza çıkmaktadır. İlerleyen bölümlerde bu yöntemleri kullanan şifreleme algoritmaları ayrıca anlatılmış ve bazı alforitmalar örnekler ile detaylandırılmıştır. Klasik şifreleme algoritmaları ve modern şifreleme algoritmalarının kullandığı şifreleme yöntemleri şu başlıklar altında toplanabilir:

1. Klasik şifreleme algoritmalarının kullandığı yöntemler
    a. Yer değiştirme (transposition) yöntemi
    b. Yerine koyma (substitution) yöntemi
        - Tekli alfabetik yerine koyma (Monoalphabetic Substitution) yöntemi
        - Çoklu alfabetik yerine koyma (Polyalphabetic Substitution) Yöntemi
2. Modern şifreleme algoritmalarının kullandığı yöntemler
    a. Blok şifeleme yöntemi
        - Yerdeğiştirme – permütasyon ağları (SPN - Substitution Permutation Network)
        - Feistel ağları
    b. Dizi/Katar (akış) şifrelme yöntemi

2.2.1. Yer değiştirme (Transposition) yöntemi

Şifrelenmek istenilen mesaja ait her harfin yerinin değiştirilmesi ve yeniden sıraya konulması ile dairesel bir sözcük oluşturulması temeline dayanan bir şifreleme tekniğidir. Bir kelimeden oluşan kısa mesajlar kısıtlı sayıda değişik sıralamalar oluşturacağı için bu gibi durumlarda elde edilecek olan güvenlik değeri oldukça düşük olacaktır. Örnek olarak sadece üç harfin oluşturduğu bir mesaj en fazla altı farklı şekilde yeniden dizilebilir, dmn, dnm, ndm, nmd, mdn, mnd. Mesaj içerisinde kullanılan karakter sayısı çoğaldıkça olası yeni dizilimlerin sayısında oldukça yüksek bir artış sağlanır, böylelikle harfleri karıştırma düzeneği bilinmeden asıl mesajı elde etme ihtimali neredeyse imkânsız bir hal alır. Bu yöntemi kullanacak olan taraflar (gönderici-alıcı) verilerin kodlanması ve kodun çözümlenmesi için kullanılacak tekniği önceden belirlemelidir.

2.2.2. Yerine koyma (Substitution) yöntemi

Yerine koyma yöntemi ile yapılacak olan şifrelemede kullanılan yöntem; şifrelenecek mesajı oluşturan harflerin yerine bir başka sembol, sayı ve ya harf yerleştirmek suretiyle gerçekleştirilir. Orijinal mesaj ardışık bitler şeklinde görülebiliyorsa, farklı bit örneklemlerinin değiştirilmesi de yerine koyma yöntemi içinde kullanılabilir. Çoklu Alfabetik Yerine Koyma (Polyalphabetic Substitution) ve Tekli Alfabetik Yerine Koyma (Monoalphabetic Substitution) yaklaşımları olmak üzere iki türlü yerine koyma yöntemi bulunmaktadır.

Tekli Alfabetik Yerine Koyma (Monoalphabetic Substitution) Yöntemi: Bu teknikte orijinal mesaj içerisinde ki her harf her defasında aynı harfle değiştirilmek suretile şifreli metin elde edilir. Bu yöntemi kullanan örnekler arasında en bilineni Sezar şifrelemesidir.

Tekli Alfabetik Yerine Koyma yöntemi kullanılanılarak oluşturulan şifreli mesajlarda harflerin yerleri değiştirilmiş olsa dahi bu harflerin kullanım sıklığını (frekansını) değiştirmez. Bu nedenle bilgisayarlar destekli sistemler tarafından çok

çabuk kırılabilirler. Tablo 2.3.'de Türkçe ve İngilizce harflerin bağıl frekans dağılımları verilmiştir. Bu tabloya bakılarak; Türkçe'de en sık kullanılan harf olan "a" harfi tablo yöntemi kullanılarak "c" harfi ile yer değiştirmesi ile ortaya çıkacak olan şifreli metinde en çok tekrar eden harfin "c" olduğu görülür ve bunun "a" harfi olabileceği tahmin edilerek şifre çözülebilir.

Tablo 2.2. Türkçe ve İngilizce harflerin bağıl frekans dağılımları [5]

Harf	Bağıl Frekans (%)		Harf	Bağıl Frekans (%)	
	EN	TR		EN	TR
A	8,2	11,92	N	6,7	4,49
B	1,5	2,84	O	7,5	2,48
C	2,8	0,96	Ö	-	0,78
Ç	-	1,15	P	1,9	0,89
D	4,3	4,70	Q	0,1	-
E	12,7	8,91	R	6,0	6,72
F	2,2	0,46	S	6,3	3,01
G	2,0	1,25	Ş	-	1,78
Ğ	-	1,13	T	9,1	3,01
H	6,1	1,21	U	2,8	3,24
I	7,0	5,11	Ü	-	1,85
İ	-	8,6	V	1,0	0,96
J	0,2	0,03	W	2,3	-
K	0,8	4,68	X	0,1	-
L	4,0	5,92	Y	2,0	3,34
M	2,4	3,75	Z	0,1	1,5

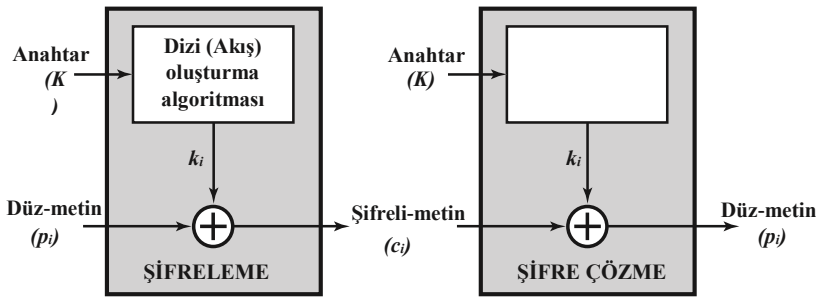
Geçmişten günümüze kadar uzanan kullandıkları alfabe ve söz dizimleri aralarında farklılıklar barındıran diğer diller gözönüne alındığında sadece bir harfe göre yapılan frekans analizleri yeterli olmayacaktır. Bunun için söz dizimlerinin ikili ve üçlü bağıl frekans dağılımlarına bakılarak daha sağlıklı bir çözümle yapılabilir. Ardı sıra gelen ve iki harften oluşan diziye digram üç harften oluşan dizilere ise trigram denir [5].

Çoklu Alfabetik Yerine Koyma (Polyalphabetic Substitution) Yöntemi: Orijinal metin içerisinden alınan çoklu harf grubu bir blok olarak kabul edilir ve bu bloklar şifreleme işlemine tabi tutulur. Böylece mesaj karakter olarak değil bloklar halinde şifrelenmeiş olur. Farklı tipteki tekli alfabetik şifreleme yöntemleri birleştirilerek, çoklu alfabetik yerine koyma tekniği oluşturulabilir. Bu şifreleme yöntemi ile mevcut dillerdeki bilinen frekans değerlerinin dışında yeni bir yapı elde edilmesi amaçlanmıştır. Bu yöntemi kullanarak şifreleme işlemini yapan algoritmaların

başında Vernam ve Playfair şifreleme teknikleri gelmektedir [5]. İlerleyen bölümlerde bu algoritmalar örnek verilerek açıklanmıştır.

2.2.3. Dizi/Katar (Akış) şifeleme yöntemi

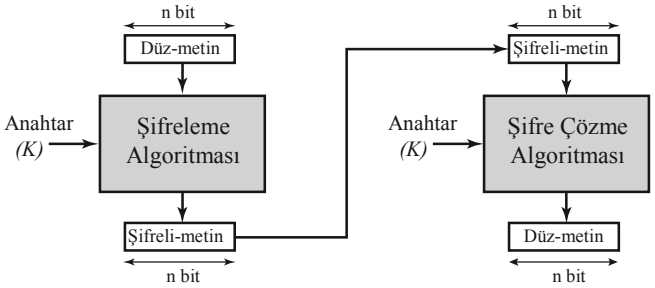
Dizi şifreleme yöntemini kullanan algoritmalar şifrelenecek veriyi bit (0 ve 1) şeklinde diziye dönüştürürler ve şifreleme işleminide bu bitler üzerinden yaparlar. Değişik uzunluğa sahip verilere uygulanabilir. Dizi şifreleme algoritmaları, şifreleme işlemi esnasında verinin belli bir uzunlukta girilip girilmemesiyle ilgilenmez. Şekil 2.2.'de gösterildiği gibi kullanıcıdan alınan anahtar üzerine uygulanacak bir algoritma ile şifreleme işleminde kullanılmak üzere bit değerlerinden oluşan dizi anahtar değerleri elde edilir. Zamana bağlı olarak üretilen anahtar dizileri sebebiyle bu algoritmalara aynı zamanda hafızalı şifreleme de denilmektedir. Gürültünün yüksek oranda mevcut olduğu telsiz haberleşme gibi ortamlarda güvenli veri iletimini sağlamak için genellikle akış şifreleme algoritmaları kullanılmaktadır.



Şekil 2.2. Dizi/Katar (Akış) şifreleme yöntemi [5]

2.2.4. Blok şifreleme yöntemi

Blok şifreleme yöntemini kullanan algoritmalar Şekil 2.3.'de gösterildiği gibi şifrelenmek istenilen mesajı bloklar şeklinde işlemektedir. Bu şifrelemeler birbirine bağlı/bağımlı olarak yapılabildiği gibi her bir blok bağımsız olarakta şifrelenebilir. Şifreleneme işlemine tabi tutulacak mesaj bloğu alınarak, üretilen ya da girilen anahtar ile şifreleme fonksiyonuna tabi tutulur. Blok şifreleme tekniklerinde dizi şifreleme tekniklerinin aksine iç hafıza bulunmadığı için bu tekniklere hafısasız şifreleme tekniği de denilmektedir.



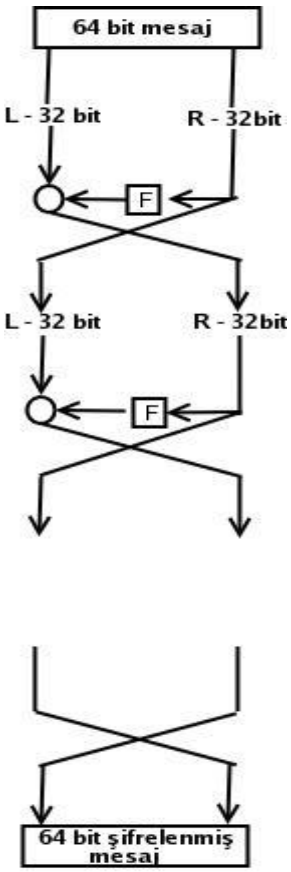
Şekil 2.3. Blok şifreleme yöntemi [5]

Blok şifreleme algoritmalarının en yaygın kullanıldığı uygulam alanları; bütünlük kontrolü gerektiren uygulamalar olarak karşımıza çıkmaktadır. Bilinen en yaygın blok şifrelemelerinin başında DES/3DES, DESX, AES, IDEA, MARS, RC6, Twofish, Blowfish, Safer algoritmaları gelir.

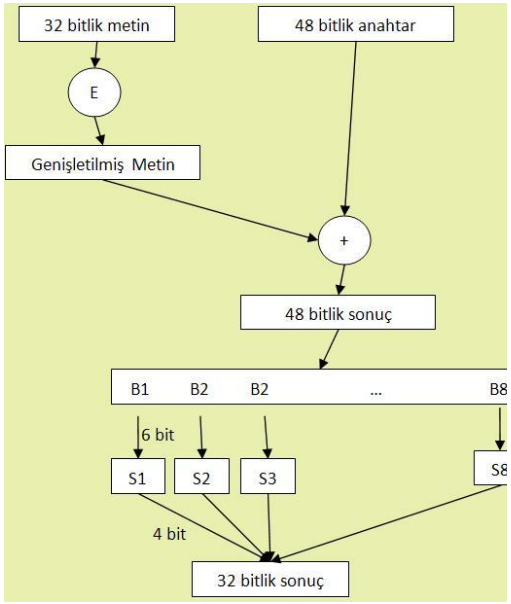
Blok şifreleme yentemi, yayılma (diffusion) ve karıştırma (confusion) ve teknikleri üzerine kurulmuş bir yöntemdir [19, 20]. Bu tekniklerden yayılma; orijinal mesajdaki izlerin kripto mesaj üzerinde algılanmamasıyla ilgilenir. Karıştırma tekniği ile hedeflenen ise orijinal mesaj ve şifreli mesaj arasındaki ilişkiyi gizlemektir. Mesaj üzerinde uygulanan yerdeğiştirme adımları ile karıştırma işlemi gerçekleştirilmiş olurken, doğrusal yer değiştirme (lineer transformation) işlemleri ile de yayılma adımı gerçekleştirilmiş olur. Feistel ağları ve Yer Değiştirme-Permütasyon ağları (SPN-Substitution Permutation Networks) olmak üzere iki ana blok şifreleme mimarisi vardır. Bu şifreleme mimarilerinin temel işleyiş yapıları sırasıyla Şekil 2.4.'de gösterilmiştir.

Blok şifreler içinde kullanılan anahtar, döngü sayısı ve S-kutuları algoritmanın gücünü belirleyen en önemli faktörlerdir.

Anahtar: Blok şifreleme algoritmalarının güvenliğini temin eden önemli etkenlerden birisi, şifrelemede kullanılacak olan anahtarın uzunluğudur. Yapılması muhtemel kaba kuvvet (brute-force) saldırısına karşı anhatar uzunluğu kritik bir önem taşımaktadır.



a) Feistel ağı şifreleme adımları



b) Feistel ağındaki F fonksiyonu

Şekil 2.4. Feistel şifreleme tekniği [17]

Döngü sayısı: Döngü sayısı şifreleme işlemi sonunda elde edilecek olan yeni yapının karmaşıklığını artırmak ve çözülebilirliğini azaltmak için büyük önem taşımaktadır. Bu sebeple algoritmada kullanılacak olan döngü sayısı iyi seçilmelidir.

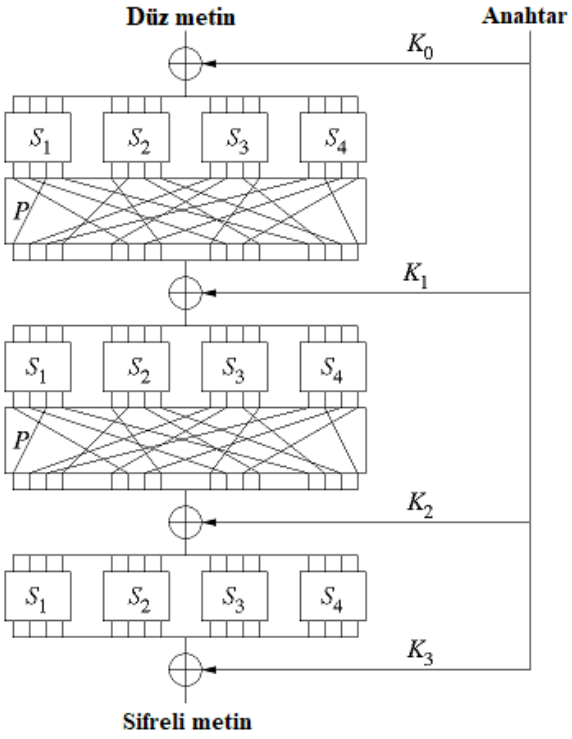
S-kutuları (Yerdeğiştirme kutuları): Algoritmanın doğrusal (lineer) olmayan tek elemanı S-kutularıdır. Algoritmada kullanılacak olan S-kutusu tercihi şifrenin karmaşıklığını doğrudan etkiler. Blok şifreleme algoritmalarının en önemli elemanın S-kutuları olduğu ifade edilmektedir [19].

Feistel ağları: Alman bilim adamı ve şifreleme uzmanı Horst Feistel tarafından tasarlanmıştır. Blok şifreleme tekniğini kullanır ve günümüzdeki diğer pek çok şifreleme algoritmasının temelini oluşturur. Feistel ağlarında şifreleme ve şifre

çözme işlemleri birbirine çok benzer hatta çoğu durumda aynıdır. Şifreleme ve şifre çözme işlemlerindeki tek fark kullanılan (oluşturulan) anahtarların kullanım sırasıdır. Feistel ağları aşağıda verilen işlem adımlarını algoritma içerisinde farklı sıralar ve miktarlarda kullanılarak oluşturulur. Bu işlem adımları şunlardır:

1. Şifrelenecek verinin bitlerinin yerinin değiştirilmesi (permütasyon şifrelemesi)
2. Basit doğrusal olmayan fonksiyon gerçeklesştirme adımı (Yerine koyma şifrelemesi, S-Kutuları)
3. Doğrusal karıştırma (yahut işlemi / özel ve ya, XOR) [17]

Yer değiştirme-permütasyon ağları(SPN): Temel olarak bir mesajın içinde bulunan harflerin bloklar halinde birbirleri ile yer değiştirmesi adımları üzerine kurulmuş bir yapıdır.



Şekil 2.5. Yer değiştirme permutasyon (SPN) ağları çalışma tekniği [17]

Bu şifreleme yönteminin tek başına kullanılması ile elde edilen şifreli yapı yeterince güvenli olmayacaktır. Bunun sebebi ise kullanılan anahtar uzunluğuna göre elde edilecek blok sayılırının tahmin edilebilir/bulunabilir olmasıdır. Şöyleki n uzunluğunda kullanılan bir anahtar ile elde edilen şifreli yapıda blok sayısı n! olacak ve şifre kırma işlemlerinin ilk adımı aşılmış olacaktır.

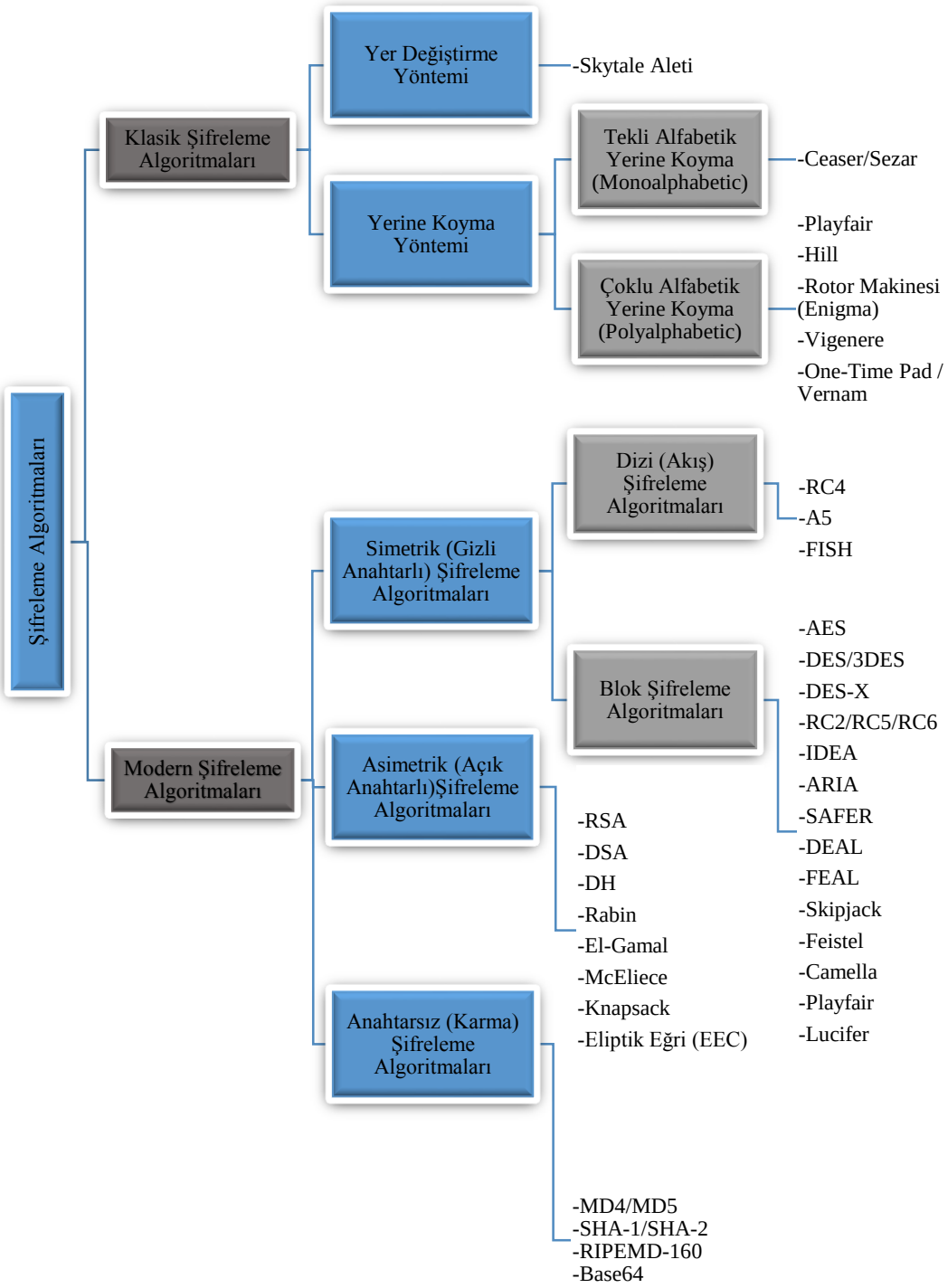
Blok şifreleme algoritmalarında şifreleme işlemleri bloklardan bağımsız veya bir birleri ile ilişkili olabileceği daha önce ifade edilmişti. Bloklar arası bu ilişkilere göre blok şifreleminin farklı türleri (mode) bulunmaktadır [5, 17]. Bunlar:

1. Elektronik kod defteri şekli (Electronic codebook mode, ECB)
2. Şifre blok zincirlemesi şekli (Cipher block chaining mode, CBC)
3. Yayılımlı şifre blok zincirlemesi (Propagating cipher clock chaining, PCBC)
4. Şifre geri beslemeli (Cipher feedback, CFB)
5. Çıktı geri beslemeli (Output feedback mode, OFB)
6. Sayıcı şekli şifreleme (Counter mode encryption, CTR, CM, ICM, SIC)

2.3. Şifreleme Algoritmaları ve Sınıflandırılması

Şifreli bir yapı elde etmek amacıyla kullanılan anahtarların uzunlukları, birbirleriyle olan ilişkisi ve anahtar türüne göre; gizli anahtarlı (simetrik) ve açık anajtarlı (asimetrik) şifreleme algoritmaları olmak üzere iki temel algoritma yapısı bulunmaktadır. Bunların dışında anahtarsız (özet) şifreleme algoritmaları ve karma şifreleme algoritmalarıda mevcuttur. Şifreleme algoritmalarının sınıflandırılması Şekil 2.6.'da gösterilmiştir.

Yapılan bu sınıflandırmanın dışında, şifreleme işleminin tersine işletilip işletilemediğine bakılarak ayrı bir sınıflandırma da yapılmaktadır. Eğer kullanılan algoritma sadece şifreleme işlemi yapıyor ise tek yönlü algoritma (örn: MD5, SHA1), hem şifreleme hemde deşifreleme (şifre çözme) adımlarını bünyesinde barındırıyor ise çift yönlü algoritma (örn: RSA, AES, DES, 3DES) olarak isimlendirilmektedir.



Şekil 2.6. Şifreleme algoritmalarının yöntem ve anahtar kullanımına göre sınıflandırılması

Bu algoritmaların kullanım yerleri; içinde barındırdıkları güvenlik seviyesi, gizlilik kapasitesi, donanımların kullanılıp kullanılmaması gibi özelliklerine göre değişikik

göstermektedir. Bu nitelikler aynı zamanda şifreleme algoritmalarının performans ölçütleri olarak da değerlendirilmektedir. Şifreleme algoritmalarının performans ölçütleri şu şekilde sıralanabilir [11, 24, 27]:

1. Algoritmanın kurulacak sisteme uygunluğu.
2. Bu algoritmaya dayalı şifreleme uygulamalarının esnekliği.
3. Uygulanan şifreleme tekniğinin kırılabilme süresi.
4. Şifreleme ve çözme işleminde ihtiyaç duyulan bellek miktarı (Bellek karmaşıklığı).
5. Şifreleme ve şifre çözme işlemlerinin ne kadar zamanda yapılabildiği (Zaman karmaşıklığı).
6. Bu uygulamaların dağıtımındaki kolaylık ya da algoritmaların standart hale getirilebilmesi.

Bu sınıflandırma içerisinde Modern şifreleme teknikleri olarak adlandırılan teknikler 19. Yüzyıl sonlarında, 20. Yüzyıl başlarında mekanik ve elektromekanik cihazların keşfedilip üretilmesiyle ortaya çıkmıştır. Bilgisayarın icat edilmesi verilerin sayısal ortamlara aktarılması ile beraber moder şifreleme teknikleri hızla gelişmiştir. Daha önceki dönemlerde ilkel yöntemler ile oluşturulan ve kullanılan şifreleme teknikleri ise Klasik şifreleme teknikleri olarak nitelendirilmektedir. Kullanılan anahtarın uzunluğu ve bünyesinde barındırdığı karmaşık şifreleme yöntemleri göz önüne alındığında; klasik şifreleme algoritmalarına nazaran modern şifreleme algoritmları daha etkin ve güvenlikli bir yapı ortaya koymaktadır.

2.3.1. Karma şifreleme algoritmaları

Sistemler içerisinde tek başına kullanılmayan, daha çok simetrik ve asimetrik algoritmalara destek amaçlı olarak kullanılan yapılardır. Güvenli şifre oluşturma, saklama ve bütünlük denetimi sağlamak amacıyla sıklıkla kullanılan karma algoritmaların en çok tercih edileni Özet Fonksiyonu (Hash Functions) adı verilen algoritmdır. Asimetirik şifreleme algoritmaları sayısal imza üretme işleminlerinda büyük yavaşlığa neden olduğundan bu tür ihtiyaçları gidermek ve yavaşlık sorununu

ortadan kaldırmak için özet fonkisyonlarını kullanmak daha doğru bir yaklaşım olacaktır. Yaygın olarak kullanılan özet fonksiyon algoritmaları DSA, SHA, MD5 olarak bilinmektedir. Ayrıca RSA'da sayısal imza doğrulamasında kullanılan diğer bir şifreleme algoritmasıdır.

2.3.2. Simetrik ve asimetrik şifreleme algoritmalarının karşılaştırılması

Bir önceki başlıklarda anlatıldığı üzere simetrik ve asimetrik şifreleme algoritmalarının kullandıkları anahtar sayısı, anahtarın uzunluğu, bünyesinde barındırdığı şifreleme yapıları, donanımla uyumlu olup olmadığı gibi etkenler bu algoritmaların önemli özelliklerini teşkil etmektedir. Bunlara bağlı olarak simetrik ve asimetrik şifreleme algoritmaları kullanım yerleri ve amaçları bakımından da farklılıklar göstermektedir. Tablo 2.3.'da bu algoritmaların genel özellikleri bu özelliklere göre karşılaştırmalarını gösteren bilgiler Tablo 2.4.'de verilmiştir.

Tablo 2.3. Simetrik ve asimetrik şifreleme algoritmalarının genel özellikleri

Simetrik şifreleme algoritmaları	**Asimetrik şifreleme algoritmaları**
- Şifreleme ve şifre çözme işlemlerinde aynı algoritma kullanılır. - İletişim yapan taraflar aynı anahtar ve aynı algoritmayı kullanır. - Gizlilik ve güvenlik ilkelerinin sağlanması algoritmanın gizli tutulmasına bağlıdır. - Elde edilen şifreli metin üzerinden kullsnılan anahtar değerine ulaşılmamalıdır.	- Şifreleme ve şifre çözme işlemlerinde kullanılan algoritma aynı fakat anahtar değerleri farklıdır. - Gönderici ve alıcının bir ortak bir de özel (gizli) anahtarı vardır. Bu anahtardan biri gizli tutulurken diğeri erişime açık olmalıdır. - Algoritma bilgisi, anahtarlardan birinin ve şifreli metin örnekleri, diğer anahtarı belirlemede yeterli olmamalı

Tablo 2.4. Simetrik ve asimetrik şifreleme algoritmalarının özelliklerinin karşılaştırılması [10]

Özellik	**Simetrik Algoritma**	**Asimetrik Algoritma**
Gizlilik	Evet	Evet
Bütünlük	-	Evet
Kimlik doğrulama	-	Evet
Performans	Oldukça Hızlı	Daha Yavaş
Güvenlik	Anahtar uzunluğuna bağlı	Anahtar uzunluğuna bağlı

Bununla beraber Ümit Günden [27] tarafından hazırlanan ve şifreleme algoritmalarının performans analizlerini gösteren çalışması konunun daha iyi analaşılmasına ışık tutacaktır.

2.3.3. Kriptografik yapıların kullanım ve uygulama alanları

Sayısal iletişimin yoğun olarak yapıldığı günümüzde, bilgi teknolojilerinin hızla gelişimi gözönüne alındığında kriptografik sistemlerin oldukça yaygın ve etkin bir şekilde kullanıldığını görmekteyiz. Son elli yıldan beri modern şifreleme yöntemleri ile oluşturulmuş ve önceki bölümlerde detaylı bir şekilde ele alınmış olan kriptografik sistemler ve bu sistemlerin en çok kullanıldığı bazı alanlar Tablo 2.5.'de listelenmiştir.

Tablo 2.5. Kriptografik sistemler ve uygulama alanları

Uygulama alanları	**Kriptogrfik sistemler**
Güvenli iletişim, Askeri haberleşme	Sim kartlar, cep telefonları
Bankacılık ve finans sektörü	Online (internet) bankacılık
Elektronik ticaret	Online alışveriş siteleri
Kimlik belirleme ve kimlik denetimi	ATM ler, bankamatik ve kredi kartları
Uzaktan erişim	Uydu alıcıları, uzaktan kumandalar,
Bilgisayar ağları ve haberleşme protokolleri	HTTPS, TCP, IP, SNMP, SMTP
Sertifika dağıtımı	SSL

2.3.4. Kriptoanaliz

Tipik olarak, bir şifreleme sistemine saldırmanın asıl amacı, şifreli (chippertext) bir metnin orijinal (plantext) haline ulaşmaktan ziyade, bu şifrelemede kullanılan anahtarı elde etmektir. Yapılacak olan (yapılması planlanan) saldırılar şifreleme işleminde kullanıldığı tahmin edilen veya öngürülen şifreleme yöntemlerine göre değiklik göstermektedir. Bu bağlamda oluşturulan kriptografik yapıların güvenlik (çözülememe) seviyeleri ayrı bir önem arz etmektedir. Genel olarak Matematiksel güvenli sistemler (Computationally Secure Systems) ve Koşulsuz güvenli sistemler (Unconditionally Secure Systems) olmak üzere iki çeşit güvenli sitem tanımı yapılmıştır [5].

Koşulsuz güvenli sistemlerde şifrelenmiş bir metin (chipper text), boyutu ne olursa olsun, şifrelenmemiş metni (orijinal düz metin) geri elde etmek için yeterli bilgiyi/ipucuyu üzerinde barındırmamalıdır. Matematiksel Güvenli Sistemlerde ise şifrelenmiş mesajı kırmanın maliyeti şifrelenmemiş mesajın değerinden fazladır. Bir

başka bakış açısıda şifrelenmiş mesajı kırmak için gereken zaman bilginin geçerlilik süresinden daha uzundur [5].

2.3.4.1. **Kripto sistemlere karşı yapılan saldırı türleri**

Geleneksel bir şifreleme şemasına saldırmak için kaba kuvvet saldırısı (brute-force attack) ve Kriptoanaliz olmak üzere iki temel yaklaşım vardır [5]. Bu yaklaşımları kullanarak şifreli metni çözmeye veya anahtarı elde etmeye yönelik girişimlerde bulunulur. Şifreli mesajlara saldırı türleri Tablo 2.1.'da gösterilmiştir.

Tablo 2.6. Şifreli mesajlarda saldırı türleri [5]

Saldırı türü	**Kripto analiz için bilinenler**
Sadece Şifrelenmiş Metin	Deşifre edilecek şifreli metin Şifreleme algoritması
Şifrelenmemiş Metin	Şifreleme algoritması Deşifre edilecek şifreli metin Bir ya da daha fazla şifrelenmemiş metin örneği çifti
Belirli Şifrelenmemiş Metin	Şifreleme algoritması Deşifre edilecek şifreli metin Şifreli metin ve bu metne karşılık gelen şifrelenmemiş metin örneği
Belirli Şifrelenmiş Metin	Şifreleme algoritması Deşifre edilecek şifreli metin Şifrelenmiş ve deşifre edilmiş anlamlı metin
Belirli Metin	Şifreleme algoritması Deşifre edilecek şifreli metin Şifrelenmiş ve deşifre edilmiş anlamlı metin Şifreli metin ve bu metne karşılık gelen şifrelenmemiş metin örneği

Kaba kuvvet saldırısı (brute-force attack): Bu yöntemde saldırgan, düz metne dönüştürülebilir bir sonuç elde edilinceye kadar şifreli bir metin parçası üzerinde mümkün olan her anahtarı çalıştırır. Ortalama olarak, başarmak için tüm olası anahtarların yarısı denenmelidir [5]. Oldukça pahalı ve uzun zaman alan bir saldırı yöntemidir. Tablo 2.7 çeşitli anahtar boyutlarına ve şifreleme algoritmalarına göre farklı hızda (saniyede bir milyar, bir trilyon) şifre çözme kabiliyetine sahip bilgisayarlar tarafından kaba kuvvet saldırısı için ne kadar sürenin gerekli olduğunu göstermektedir.

Tablo 2.7. Kaba kuvvet saldırılarında şifre kırmak için gereken ortalama süreler [5]

Anahtar Uzunluğu (bit)	**Şifreleme Algoritması**	**Oluşturabilecek Anahtar Sayısı (2^n)**	**10^9 şifre/s hızında ortalama çözme süresi**	**10^{13} şifre/s hızında ortalama çözme süresi**
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns ≈ 1 yıl	1 saat
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	$2^{127} = 5.3 \times 10^{21}$ yıl	5.3×10^{17} yıl
168	3DES	$2^{168} \approx 3.7 \times 10^{50}$	$2^{167} = 5.8 \times 10^{33}$ yıl	5.8×10^{29} yıl
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	$2^{191} = 9.8 \times 10^{40}$ yıl	9.8×10^{36} yıl
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	$2^{255} = 1.8 \times 10^{60}$ yıl	1.8×10^{56} yıl
26 karakter (permutasyonu)	Tekli yerine koyma (Monoalphabetic)	$2! = 4 \times 1026$	$2 \times 10^{26} = 6.3 \times 10^{9}$ yıl	6.3×10^{6} yıl

BÖLÜM 3. STEGANOGRAFİ

Steganografi veri güvenliği ve gizliliğini sağlamak için geliştirilmiş olan başka bir yazma tekniğidir. Kriptografi, iletilmek istenen bir veriyi şifreleyerek onun çözülebilirliğini ve anlaşılabilirliğini engellemeyi amaçlarken, steganografi ise mevcut bilginin görünmezliği ile uğraşır [1, 29]. Kriptografik ve steganografik yapıların özellikleri ve bu yapıların arasındaki temel farklar bu bölüm içerisinde alt başlık olarak bölüm 3.3.'de anlatılmıştır. Steganografi, bir verinin başka bir veri içerisine saklanarak/gömülerek alıcı tarafa iletilme sanatıdır. Steganografi şifrelemenin alternatifi değil onun tamamlayıcısıdır. Yakın zamanda ve günümüzde bu iki (kriptografi, steganografi) tekniğin birarada kullanan hibrit örnekler artmaktadır [25, 26, 61, 42]. Yapılan bu tez çalışmasında da şifreleme (krptografi) ve veri gizleme (steganografi) teknikleri beraber kullanılmıştır. Bilgi gizleme sanatı başlığı altında yapılan diğer bir yöntem ise Damgalama/Filigranlama işlemleridir. Filigranlama teknikleri genellikle ortama/nesneye ilişkin bilgiler ile (telif hakkı gibi) ilişkilidir [5, 30]. Steganografi işlemleri kapasite, algılanamazlık ve güvenlik temelleri üzerine kurulu bir yapı iken, filigranlama ise yapılış amacına göre daha çok sağlamlık temelleri üzerine kurulmuştur. Bu iki yapı arasındaki farklar Bölüm 3.4.'de anlatılmıştır. Steganaliz ise taşıyıcı nesne içinde başka bir veri olup olmadığını anlamaya yarayan analiz teknikleridir. Yine bu konu da steganaliz başlığı altında ayrıca anlatılmıştır.

3.1. Steganografi Tarihçesi

Steganografi, taşıyıcı bir ortam üzerinde herhangi bir gizli mesajın varlığını gizleyerek iki kişinin gizlice iletişim kurmasına izin veren bir tekniktir. Bu yeni bir terim ya da teknik değil, geçmişi uzun yıllar öncesine dayanan bir veri gizleme sanatı olarak karşımıza çıkmaktadır. "Görünmeyen" anlamına gelen Latincedeki 'steganos' kelimesi, steganografi biliminde 'gizlenmiş-örtülü yazı' anlamına gelmektedir [4].

Günlük yaşamımızda da steganografinin parçası olduğunun farkında olmadan gizlice farklı yollarla iletişim kurarız.

Geçmiş dönemlerde eldeki imkân ve materyaller doğrultusunda güvenli ve gizli iletişim sağlayabilmek için çeşitli yöntemler geliştirilmiş ve kullanılmıştır. Bu dönemde gerçekleştirilen veri gizleme çalışmaları Geleneksel Steganografi olarak adlandırılmaktadır.

Heredot hikâyelerinde de yer alan Persler ile Yunanlılar arasında yapılan savaşta, Yunan imparator Histiaeus'un kölesinin saçını kazıtıp gizli mesajı dövme şeklinde kafa derisine işler, köle Milet'e vardığında saçları tekrar kazıtılarak kafa derisinden imparatorun gönderdiği gizli mesajı içeren dövme ortaya çıkar [1, 4]. Bu uygulama tarihte bilinen ilk steganografi yaklaşımlarındandır.

Romalılar döneminde gizli haberleşme yapabilmek adına görünmez mürekkepler kullanmıştır. Rönesans döneminde ise akrostiş şiirler yazılmış bu şiirlerin baş harfleri birleştirilerek gizli iletişim yapılmıştır [1, 4].

Geçmiş dönemlerde ağırlıklı olarak özel mürekkep ve kimyasal maddeler kullanmak suretiyle görünmez yazılar üzerinden gizli haberleşmeler yapılmıştır. Yine bu dönemlerde metin steganografisi örneklerine başta şiirler olmak üzere birçok yazılı metinde rastlamak mümkündür [1, 4]. Çinlilerin kullamış oldukları meyve sepetleri de steganografik yaklaşım örneklerindendir. Bu sepetlerdeki meyvelerin birbirlerine göre konumları farklı anlamlar ifade etmekteydi. Böylece başkaları anlamadan kendi aralarında gizli bir iletişim kurabilmişlerdir.

Almanlar 20 inci yüzyılda yaşanan II. Dünya savaşında geliştirdikleri mikro-noktalama aletini kullanmışlardır. Geliştirelen bu alet ile gizlenmek istenilen mesaj, resimleme tekniği kullanılarak harflerin veya noktalama işaretlerinin arasına yerleştrilmitir. Karşı taraf mesajı çözmek için oluşturulan (resmedilen) noktalama işaretlerini birleştirmesi yeterli olacaktır [32].

Sayısal ses dosyaları üzerinde yapılan ilk steganografi işlemi 1954 yılında gerçekleştirilen, müzik dosyalarına sahiplik bilgisinin eklendiği fligranlama işlemidir. Bu çalışmadan sonra ses ve müzik dosyaları üzerine/içerisine veri gizleme işlmeleri giderek artmıştır.

1960 lı yıllara gelindiğinde mor ötesi boya ile yazı yazabilen sprey ve kalemler steganografi işlemlerinde etkin bir şekilde kullanılmıştır. Bu kalemlerin yazdığı yazıların özelliği sadece mor ötesi ışıkla görülebiliyor olmasıdır.

İletişim çağı olarak adlandırılan günümüzde ise steganografi uygulamaları daha çok sayısal nesneler üzerinde gerçekleştirilmeye başlanmıştır. Sayısal ortamlar üzerine yapılan veri gizleme işlemlerinin yanı sıra bu gizleme işlemlerine karşı gerçekleştirilen veri çıkarma (steganaliz) çalışmaları da giderek artmaktadır. Steganaliz teknikleri ve yöntemleri geliştikçe buna bağlı olarak veriyi saklama yöntemleri de zamanla gelişmiştir.

Önceki paragraflarda anlatıldığı gibi Almanların geliştirdiği ve II Dünya savaşında kullandığı mikro noktalama tekniği, akabinde 1960'lı yıllarda müzik dosyaları üzerinde gerçekleştirilen filigranlama işleminden sonra en çarpıcı steganografik gelişme ABD'li bilim adamları tarafndan 1999 yılında DNA içinde organik moleküllerin yerlerini değiştirerek gizli veri saklama işleminin yapılmasıdır. Böylece çok büyük kapasiteye sahip veriler küçük bir DNA parçası içinde gizlenilebilir olmuştur [32].

Bilgisayar teknolojisinin ilerlemesi sonucunda işletim sistemleri içerisinde veri gizlenmeye ve bu yollarla veriler taşınmaya başlandı. Windows işletim sisteminin ilk çıktığı yıllarda 1 KB'lık verinin işlem görmesi sonrasında 32 KB'lık bir alan ayrıldığını fark eden uzmanlar bu fazladan tahsis edilen alanlara veriyi gizlemeye basladı. Buna benzer bir stegangrafik yaklaşım TCP/IP paketleri üzerinde gerçekleştirilmiştir. Bu gelişmenin ardından steganografi günümüzde çok büyük bir gelişme göstermistir. Bilgisayar ve iletişim teknolojisinin ilerlemesiyle beraber yeni bir düzen kazanan steganografi günümüzde yalnızca metin değil, ses, resim ve video

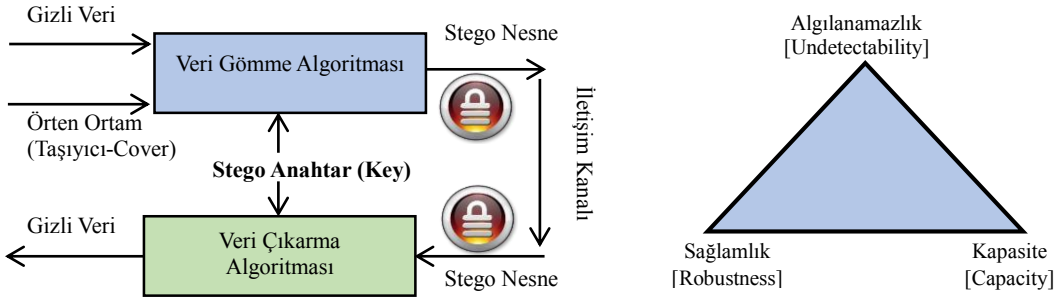
dosyaları başta olmak üzere birçok sayısal formda 1 ve 0 olarak kodlanarak uygulama alanı bulmuştur [32]. İnternet ve intranet üzerinden yapılan haberleşme ve iletişimin artması sonucu ağ paketleri ve protokller (TCP/IP, UDP, SCTP) veri gizleme işlemleri için yeni bir alan ortaya çıkarmıştır [62, 63].

Steganografi bilimi, kendi içerisinde ve Teknik Steganografi Dilbilim Steganografi olmak üzere ikiye ayrılmaktadır. Bu konuda Bölüm 3.5.'de Bilgi gizleme ve steganografinin sınıflandırılması başlığında detaylandırılmıştır.

3.2. Steganografi Temelleri ve Gereksinimleri

Steganografide, gönderenin başka bir dosya veya nesne altındaki mesajı kapatarak alıcıya bazı gizli bilgileri bir iletişim kanalı üzerinden göndermesi gerekir. Bilgi saklama bir iletinin, başka bir veri kaynağı içerisinde (metin, resim, ses, video v.s.) değişik yöntemler veya teknikler kullanarak gizlenip alıcı tarafa iletilmesini hedeflemektedir [30, 31]. Bu bakımdan steganografik yaklaşımlarda mesajın gömüleceği/gizleneceği taşıyıcı nesne (cover object) ve bu nesnenin veri saklama kapasitesi büyük önem taşımaktadır. Gizlenecek olan veri genellikle bir anahtar (key) ilave edilerek veri gömme algoritmaları kullanılarak taşıyıcı nesne içerisne gömülür.

Bu şekilde elde edilen, içerisinde taşıyıcı nesne ve gizli veriyide barındıran yeni nesne stego nesne halini alır. Şekil 3.1.'de steganografi özellikleri ve bir stegosistem deki temel bileşenler gösterilmiştir. Oluşturulacak olan stego nesne içersinde yeteri kadar veri taşıma kapasitesini barındırmalıdır. Öyleki eklenecek olan gizli mesaj sonrası taşıyıcı nesnede farkedilebilir bir değişiklik olmasın. Bu steganografinin algılanamazlık ilkesini doğrudan ilgilendirir. Kod çözme veya veri çıkarma işlemi ise taşıyıcı nesne ve kullanılan anahtarı girdi olarak alır ver veri çıkarma algoritması uygulanarak gizli mesaj geri elde edilir.



a) Stegosistem yapısı ve temel bileşenler b) Steganografi gereksinimleri [35]

Şekil 3.1. Stegosistem yapısı, temel bileşenleri ve gereksinimleri

Steganografik bir yapı incelenirken göz önüne alınan 3 temel unsur vardır [63]:

- Algılanamazlık (Değişimin fark edilememesi)
- Kapasite (Taşıyıcı nesne üzerinde saklanabilecek veri miktarı)
- Sağlamlık / Dayanıklılık

Sağlıklı ve etkin bir steganografik sistem steganaliz yöntemlerine karşı koyabilmek için baraj grültüsü ve şifreli güvenlik kavramlarını gözönüne alarak hazırlanmalıdır.

Baraj gürültüsü: Stego nesneye ait sinyalde, gizli veri eklenmesi haricinde oluşan, kodlama, iletim, kayıt, vb. nedenlerden meydana gelen bozulma olarak tanımlanır. Steganografik bir yapıda gizli verinin varlığının tespit edilmesi (pasif steganaliz) ve gizli verinin içeriğinin açığa çıkartılması (aktif steganaliz) zor olması isteniyorsa, gizli veri gömme işlemi sırasında oluşan bozulma sinyali baraj gürültüsünün altında kalmalı, dahası tüm istatistiksel özellikleri baraj gürültüsü ile aynı, çok yakın veya benzer tutulmalıdır. Bu da steganografi işlemlerinin üç temel ilkelerinden olan algınalamazlık ve sağlamlık ilkelerinin başarıyla yerine getirlmiş olması demektir.

Şifreli gizli veriler: İletilmek istenilen orijinal veriler taşıyıcı ortama gömülmeden önce şifrelenirler. Mesajın şifrelenmesi aktif steganaliz yöntemlerine karşı kesin çözüm sağlar; kriptogafik teknikler ile şifrelenmiş olan gizli veri gömülü olduğu sinyalden bir şekilde soyutlanıp çıkarılması durumunda, veri içeriğinin elde edilmesi için zorlu ve karmaşık kriptoanaliz sürecinin başarıyla sonuçlandırılması gerekir.

Dayanıklılık taşıyıcı nesne içersini gömülen mesajın saldırılar sonucunda bile, elde edilememesi ilkesine dayanmaktadır. Steganografi işlemlerinde kullanılmak üzere geliştirilmiş olan veri gömme algoritmaları bütün saldırılara karşı tam manasıyla bir koruma sağlamamaktadır. Bu da tek başına steganografi yaklaşımının istenilen gizliliği sağlayamayacağı anlamına gelmektedir.

Yapılacak olan veri gizleme çalışmasının amacına göre (filigranlama/steganografi) Şekil 3.1.'de gösterilen temel gereksinimler arasında karşılıklı olarak bir ödünleşim meydana gelmektedir. Yani daha fazla veri saklanmak istendiği zaman kapasite ve algılanaazlık ilkeleri arasında bir ödünleşim olurken, daha etkin bir güvenlik sağlanmak istendiğinde ise güvenlik ve kapasite ilkeleri arasında bir ödünleşim söz konusu olabilir. Steganografik yaklaşımlarda kapasite ve güvenlik ilkeleri ön plana çıkarken, filigranlama işlemlerinde güvenlik ve dayanıklılık ilkeleri ön plandadır.

3.3. Kriptografi ve Steganografi Arasındaki Farklar

Kriptografi bir iletinin/verinin okunabilirliğini engellemek amacıyla farklı teknik ve yöntemler kullanarak o verinin şifrelenmesini amaçlar. Verinin görünür olup olmamasıyla ilgilenmez, ana amaç mesaj anlaşılamaz hale getirmektir. Steganografi ise iletinin fark edilmeden karşı tarafa gönderilmesi ilkesine dayanmaktadır. Bunun içinde mesajın taşınacağı bir haberleşme kanalı, mesajı barındıran bir taşıyıcı, mesajın hangi taşıyıcı ile ve hangi yolla gönderileceği bilgisi ve mesajın nasıl çözüleceği bilgisi önem taşımaktadır. Steganografinin kriptografiye göre en büyük avantajı, insan duyularını kullanarak bilgiyi gören/duyan bir kimsenin, iletilenin içinde gizli bir mesaj olup olmadığını bilemiyecek olmasıdır. Kriptografi ve Steganografi tekniklerinin işlevleri ve aralarındaki farklar Tablo 3.1.'de gösterilmiştir.

Tablo 3.1. Steganografi ve kriptografi arasındaki farklar.

İşlem/Özellik	Kriptografi	Steganografi
Bilgiyi üçüncü tarafların anlaşılamayacağı şekilde dönüştürmek	Evet	Evet
Bilgiyi gizleme	Hayır	Evet
Anahtar (Key) kullanımı	Evet	Evet
Gizli iletişim olgusunu göstermemek/gizlemek	Hayır	Evet
İletişim yapan tarafların anonimliğini/bilinmezliğini sağlama	Hayır	Evet
Ek taşıyıcı ihtiyacı	Hayır	Evet
İletişim sürecinde aktarılan bilgi miktarı	Şifrelenmiş bilgi miktarıyla karşılaştırılabilir	Taşıyıcı ortamın kapasitesine ve kullanılan teknipe bağlı olmakla beraber, şifrelenmiş bilgi miktarından çok daha büyüktür.

3.4. Filigranlama/Damgalama ve Steganografi Arasındaki Farklar

Steganografi ve filgranlama tekniklerinin her ikiside bilgi gizleme sanatının içinde yer alan yöntemlerdir. Bu iki tekniğin uygulama yerleri benzer olsada, kullanım amaçları ve uygulanma şekilleri farklıdır. Steganografide gizlenecek olan bilgi taşıyıcı/örten ortam içerisine görünmez/algılanamaz bir şekilde yerleştirilmelidir. Yine bu teknikte gizlenecek olan veri taşıyıcı ortam ile herhangi bir ilişki barındırmak zorunda değildir. Yani taşıyıcı ortamdan bağımsızdır. Filigranlama işleminde ise nesne (taşıyıcı ortam) üzerine/içine yerleştirilen bilgi bu nesne ile alakalıdır [30]. Filigranlama işlemlerinde görünmezlik/algılanamazlık herzaman geçerli değildir, kullanım amacına göre görünür veya görünmez olabilir. Örnek olarak televizyon yayınlarındaki logo gösterimini göz önüne alacak olursak, burda kullanılan logo gizli değildir ve ekranda görünür durumdadır [41]. Müzik dosyaları içerisine yerleştirilen sahip bilgiside (telif hakkı) damgalama işlemi olarak adlandırılır, bu ve benzeri damgalama işlemleri gömülü/görünmez filigran olarak adlandırılırlar. Steganografik yaklaşımlara en çok benzeyen filigran yöntemleri gömülü damgalama yöntemleridir. Filigranlama da gizlilikten daha çok güvenlik ön plandadır. Buradaki amaç taşıyıcı ortam üzerine/içine yerleştirilen bilgilerin ortamdan çıkarılamamsını/ayrıştırılamamasını hedefler [34]. Steganografide ise

algılanamazlık /gizlilik sağlamlık ve veri gömme kapasitesi kadar önemlidir. Ayrıca filigranlama işleminde yerleştirilecek olan bilginin boyutu kısıtlı olmalıdır.

3.5. Steganografide Kullanılan Temel Teknikler

Steganografi uygulamalarında gizlenmek istenilen veri dışardan alınan bir anahtar ilavesiyle etkin bir gizleme algoritması kullanıp taşıyıcı bir nesne içerisinine gömülmesi hedeflenir. Bu amaç doğrultusunda taşıyıcı nesnenin türü ve niteliği büyük bir önem taşımaktadır. Taşıyıcı nesnenin türüne göre değişiklik göstermekle beraber, veri gizleme işlemi taşıyıcı nesneye uygulanma yeri ve şekli açısından temelde üç farklı tekniğe/yaklaşıma göre gerçekleştirilmektedir. Bu teknikler;

1. Uzamsal/Geçici Alan Teknikleri. (Temporal-Spatial Domain v)
    a. Düşük bit (LSB) kodlama
    b. Eşlik (Parity) kodlama
    c. Yankı (Echo) kodlama
2. Dönüştürme Alan Teknikleri. (Transform Domain Techniques)
    a. Tayf yayılması (Spread Spectrum)
    b. Faz/Aşama kodlaması (Phase Coding)
    c. Ayrık dalgacak dönüşümü (Discrete Wavelet Transform - DWT)
    d. Ayrık kosinus dönüşümü (Discrete Cosine Transform - DCT)
    e. Ayrık Fourier dönüşümü (Discrete Fourier Transform - DFT)
    f. Gürültü/Ton ekleme (Tone Insertion)
    g. Genlik kodlama (Amplitude Coding)
3. Sıkıştırılmış Alan Teknikleri (Compressed Domain Techniques)
    a. Vektör miktarı (Vector Quantization)
    b. Fraktal sıkıştırma (Fractal Compression)

Olarak karşımıza çıkmaktadır. Aşağıda kullanılan bu teknikler hakkında özet bilgi verilmiş, bu teknikler altında yeralan alt yöntemlerin bazıları ise ses steganografi teknikleri başlığı altında anlatılmıştır.

3.5.1. Uzamsal alan teknikleri

Bu yöntem yüksek gömme kapasitesine sahip ve kolay uygulanabilse de, düşük sağlamlığa sahiptir. Genellikle en düşük anlamlı bitler üzerinde yapılan steganografi yaklaşımı üzerine kuruludur. Diğer yaklaşımlara oranla daha az güvenlik ve sıkıştırmaya duyarlılık gibi bazı dezavantajları vardır. Saldırgan tüm LSB bitlerini toplayarak gizli mesajı kolayca ortaya çıkarabilir veya LSB bitlerini yeniden kodlayabilir. Pek çok teknik sağlamlığı artırmak için uzlamsal alan üzerinde gerçekleştirilen steganografiyi diğer yöntemlerle birleştirmeye çalışır.

3.5.2. Dönüşüm alan teknikleri

Veri gizlemede kullanılan diğer bir etki alanı da dönüşüm alanı teknikleridir. Bu yaklaşımda, ilk önce kapak dosyası dönüştürülür ve daha sonra gizlenecek veri dönüşüm katsayıları içerisine gömülür. Yine bu yaklaşımda steganografi sisteminin verileri algısal olarak önemli bileşenlere gömmesini sağlar ve gömülü verilerin açığa çıkartılmasını zorlaştırır. Böylece amplifikasyon ve filtreleme gibi sinyal bozmalarına karşı yüksek güvenlik ve sağlamlık sunmaktadır. Öte yandan, gizlenecek veriler üzerinde kodlama esnasında herhangi bir sıkıştırma yapmak mümkün değildir. Steganografide kullanılan en yaygın dönüşümler Ayrık dalgacık dönüşümü (DWT), Ayrık Kosinüs Dönüşümü (DCT), Ayrık Fourier Dönüşümü (DFT) 'dir.

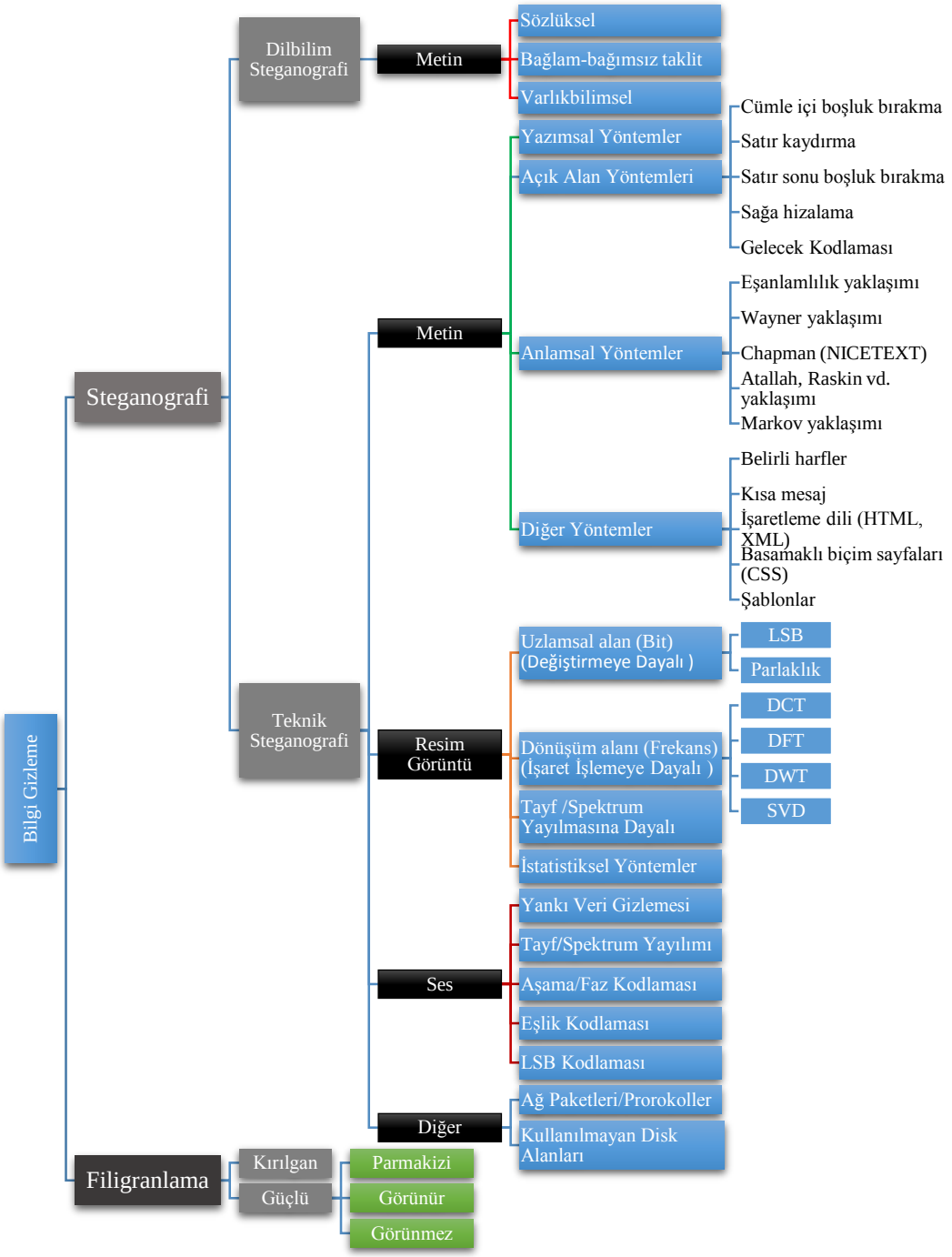
3.5.3. Sıkıştırılmış alan teknikleri

Bu alanda örtü/taşıyıcı verileri veya stego nesneler, steganografi teknikleri geliştirmek ve yüksek kapasite ve sıkıştırma oranı üretmek için farklı sıkıştırma teknikleri kullanılarak sıkıştırılmaktadır. Vektör miktarı (VQ), gizli verileri sıkıştırılmış kapak dosyasında gizlemek için kullanılan bir tekniktir. Fraktal sıkıştırma (FC), gizli verileri kapak dosyasına gizlemeden önce sıkıştıran başka bir tekniktir. Bunlar, kullanılan en yaygın sıkıştırma teknikleridir.

3.6. Steganografi Sınıflandırılması

Bilgi güvenliğini sağlayan iki temel yaklaşım vardır. Bunlardan biri veri şifreleme (kriptografi) bir diğeri ise bilgi gizleme (information hidding) yaklaşımıdır. Steganografi ve filigranlama işlemleri bilgi gizleme yaklaşımının altında yer alan güvenli yazma teknikleridir. Önceki başlıklarda steganografi ve filigranlamanın arasındaki farklar anlatılmıştı.

Steganografik yaklaşımlar kendi içerisinde Teknik Steganografi ve Dilbilim Steganografi olmak üzere ikiye ayrılmıştır. Dilbilim steganografisinde taşıyıcı ortam olarak metin (text) verileri kullanılır. Bu yapıda bir yaklaşım ile yapılacak olan bilgi gizleme oldukça zor ve kısıtlıdır ve kullanılan dile bağlıdır. Bilgisayar tabanlı yöntemler kullanarak sayısal veriler (metin, ses, resim, video) üzerinde daha etkin bir veri gizleme yapmamıza olanak sağlayan yapı ise Teknik Steganografi olarak isimlendirilmştir. Bilgi gizleme yöntemlerinin detaylı bir şekilde sınıflandırılması Şekil 3.2.'de gösterilmiştir. Yapılan bu sınıflandırmadan da görüleceği üzere steganografi işlemleri günümüzde ağırlıklı olarak sayısal veriler üzerinde yapılmaktadır. Bu ortamlar steganografinin uygulanma alanları olarak nitelendirilebir. Bunların dışında ağ paketleri üzerinde, disklerin kullanılmayan alanları üzerinde de steganografi uygulamaları vardır. Daha önceki yapılan akademik ve bilimsel çalışmalara bakıldığında metin (text), resim (image) steganografisi üzerinde oldukça fazla durulmuştur. Görüntü (video) steganografiside resim steganografisinin bir türevi olarak benzer şekilde uygulanmaktadır. Bu bakımdan bundan sonraki alt başlıklarda metin ve resim steganografileri yüzeysel olarak anlatılmış, bu çalışmada da kullanılan ses (audio) steganografisi ise kullanılan yöntemler ile birlikte detaylandırılmıştır.



Şekil 3.2. Steganografinin uygulama ortamları, kullanılan yöntem ve tekniklere göre sınıflandırılması

3.6.1. Metin (Text) steganografi

Taşıyıcı ortam olarak metinlerin kullanıldığı ve uygulanması zor bir veri gizleme şekli olan bu steganografik yaklaşımda gizlenecek veri oranı oldukça kısıtlıdır. Taşıyıcı olarak kullanılan metnin'in içindeki gereksiz/kullanılmayan alanların ve

boşlukların miktarının az olması ve insanların farklı görünümdeki metinlere karşı daha duyarlı olmasından dolayı veri gizleme oranı düşüktür [30]. Taşıyıcı nesne olarak metin kullanan steganografik yaklaşımlar, gizli mesajı geri elde edebilmek için orijinal metnin biçimlendirme bilgisine ve ya orijinal metnin kendisine ihtiyaç duymaktadır. Metin Steganografisi uygulunan tekniklere veya veri saklanacak yerlerin özelliklerine göre birçok farklı şekilde gerçekleştirilebilir. Bu yöntem ve kullanılan tekniklerin başlıkları Şekil 3.2.'de gösterilmiştir.

Metin içinde gizlenen veriler, telif hakkı doğrulama, kimlik doğrulama ve ek açıklama dâhil olmak üzere çeşitli uygulamalara sahiptir. Metinden ayrılmaz olan telif hakkı bilgilerinin yayınlanması, yayıncıların ürünlerini elektronik dağıtımın artması çağında korumaları için bir yoldur. Ek açıklama sabotaj koruması için kullanılabilir. Örneğin, kâğıdın bir şifreleme karesi kâğıda kodlanmışsa, dosyanın değiştirilip değiştirilmediğini belirlemek basit bir meseledir. Doğrulama, bir sunucu tarafından kolayca gerçekleştirilebilecek görevler arasındadır; bu durumda, uygun olduğu şekilde "otantik" veya "doğrulanmamış" kararını iade edecektir [45].

Metin steganografisi kullanılan bu tekniklerden bazılarını örnek vererek açıklayacak olursak; İkinci dünya savaşında bir alman casus tarafından kullanılan yöntem orjinal metindeki ikinci harflerin alınarak birleştirilmesine dayanıyor. Bu yöntem metindeki belirli harfleri kullanma tekniğine dayanarak oluşturlan steganografik bir yapıdır. Gönderdiği mesaj:

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

Bu mesajın her kelimesindeki ikinci harfleri alıp birleştirdiğimizde karşımıza iletilmek istenen asıl gizli mesaj:

"Pershing sails from NY June 1" [33]. Olarak çıkmaktadır.

Günümüz teknolojisi gözönüne alındığında metinsel ifade oluşturma şekilleri, metinleri biçimleyen ve niteleyen yeni metinsel teknikler ortaya çıktığını görürüz. Bunların başında HTML (Hyper Text Markup Language – Hareketli Metin İşaretleme Dili), XML (Extensible Markup Language - Genişleyebilir İşaretleme Dili), CSS (Cascading Style Sheets - Basamaklı Stil Şablonları) gibi metin işleme ve biçimlendirme yapıları gelmektedir. Bu yapılar içersinde de steganografik çalışmalar yapılmış ve kullanılmıştır. HTML ve XML gibi etiket (tag) tabanlı işaretleme dillerinde, etiketlerin kapanış biçimleri, etiketlerin büyük harf veya küçük harfler ile ifade edilmesi gibi yaklaşımlar veri gizleme için kullanılan basit ve etkin bir yöntem olarak karşımıza çıkmaktadır [30, 34]. Örnek olarak <p align="center">, <p align="cenTER">, <p align="Center"> ve <p aLigN="center"> etiketlerinin hepsi aynı biçimlemeyi işaret etmektedir. Bilindiği gibi HTML etiketlerinde, etiket içerisindeki özellik niteleyicilerinde büyük/küçük harf duyarlılığı yoktur. Burdan yola çıkarak etiket içerinde kullanılan ifadelerde büyük harfler 1, küçük harfler 0 olarak kabul edilerek veri gizleme gerçekleştirilebilir. Veri gizleme kapasitesini artırmak için etiketler bir bütün olarak değil, etiketin her bir harfi ayrı ayrı ele alınabilir ve böylece muazzam bir veri saklama kapasitesi elde edilebilir. Buna ek olarak etiketler ve içerisindeki değerlere boşluk ekleme gibi diğer metinsel steganografi yöntemleride burda ilave olarak kullanılabilir. HTML etiketleri üzerinde uygulanabilecek olan başka bir yaklaşımda etiketlerin kapanış biçimleri olabilir. Örnek olarak; <img> </img> yapısı ile <img/> yapıları sonuç olarak aynı nitelemeyi ifade etmektedir. İlk ifade biçimini <img> </img> 1, ikinci ifade biçimini de <img/> 0 olarak kabul ederek veri gizleme işlemi yapılabilir. Anlatılan bu yaklaşım tarzlarını göz önüne alarak "DUMAN" versinin kodlanış biçimi Tablo 3.2.'de gösterilirken, HTML etiketleri ile gizleyen örnek şekil 3.3.'de verilmiştir.

Tablo 3.2. DUMAN Versinin kodlanması

Gizli Veri	ASCII Karşılığı	İkili (Binary) Karşılığı
D	068	01000100
U	085	01010101
M	077	01001101
A	065	01000001
N	077	01001110

```
<hTml>
        <hEad>
                <tItLe> HTML STEGO </TiTlE>
        </heAD>
        <bOdY>
                <img src=r1.jpg/>
                <img src=r2.jpg/>
                <img src=r3.jpg/>
                <img src=r4.jpg/>
                <img src=r5.jpg/>
                <img src=r6.jpg> </img>
        </bOdy>
</HTMl>
```

Şekil 3.3. DUMAN Verisinin HTML etiketleri içerisine gizlenmesi

HTML etiketlerinin biçimleri ve bunlara karşılık gelecek olan değerler şu şekilde ele alınmıştır:

1. Etiket içeriğine göre büyük harfler 1, küçük harfler 0
2. Etiket açılış kapanış şekillerine göre <img> </img> 1, <img/> 0 olarak kabul edildi

Kodlamada kullanılan birinci (büyük/küçük etiket harfi) teknik, saklama kapasitesini artıtırken, yapılacak olan görsel ataklara karşı stego ortamın algılanamazlık ilkesini olumsuz yönde etkileyecektir.

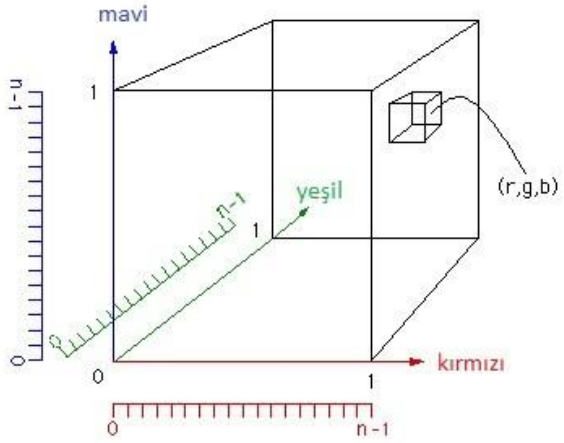
Bu örnekte olduğu gibi HTML, XML etiketleri üzerinde daha farklı şekillerde fonksiyonlar üretilerek veri gizleme için alternatifler oluşturulabilir. Bunun yanında örnekteki yaklaşım veya diğer metin steganografi tekniklerin en büyük dezavantajı ise incelemeye alındıklarında paragraf şekilleri, hizalama şekilleri, cümle/kelime arasındaki boşluk sayıları, etiketlerin büyük/küçük harfler ile yazılması, etiketlerin açılış/kapanış şekilleri ve diğer birçok yaklaşımın kolayca bir şekilde fark edilmesidir. Etkin bir steganaliz çalışmasına ihtiyaç duymadan, herhangi bir metin editörü ile yapılabilecek olan düzenleme veya düzeltmeler karşısında dayanıksız ve korunmasızdırlar [34].

3.6.2. Resim (İmage) steganografi

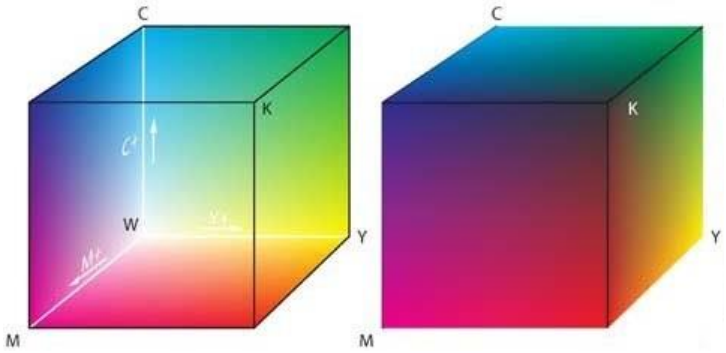
Bu yaklaşımda ise taşıyıcı ortam olarak sayısal resim dosyaları kullanılmaktadır. Bu bağlamda taşıyıcı nesne olarak kullanılan bir çok resim dosyası türü (BMP, GIF, PNG, JPEG, TIFF vb.) vardır. Bu resim dosyaları içerisinde steganografi uygulamaları için en uygun olanı 24 bit formattaki BMP dosyalarıdır. Bu dosyalar üzerinde herhangi bir sıkıştırma işlemi uygulanmadan resimler oluşturulduğu için veri gizlemek için yeteri miktarda alan vardır.

Sayısal resim dosyaları pixel ismi verilen hücrelerden oluştrulmuş olan ve herbir pixel içerisinde de bir veya birden fazla rengi barıdıran bir yapıya sahiptir. Bu dosyalar içerisindeki renkleri sınıflandırmak, ayrıştırmak ve de standartlaştırmak için renk uzayı kavramı ortaya çıkmıştır [36, 37]. Ortaya çıkan bu renk uzaylarının (renkleri tanımlama ve ayrıştıma) başında RGB (Red-Kırımızı, Green-Yeşil, Blue-Mavi), CMYK (Cyan, Magenta, Yellow, Key), HSV (Hue, Saturation, Value – Renk Özü), YUV (Y Luminance, U Chrominance1, V Chrominance2) renk uzayları gelmektedir. Bu renk uzayları sayısal görüntü işlemlerinde önemli bir yeri tutmaktadır.

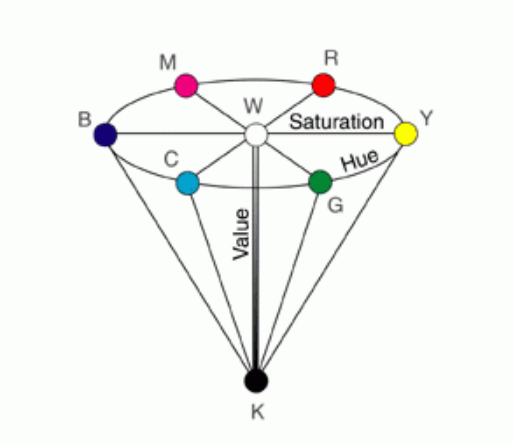
Her renk uzayı, Şekil 3.4.'de gösterildiği gibi kendine özgü bir yapıyla renk kümesini tanımlamaktadır. Siyah beyaz bir görüntü sadece 2 adet değişkene sahip olduğundan dolayı bu görüntüleri dijitalleştirmek çok daha kolaydır. Renkli bir görüntüyü sayısal ortama aktarmak için sadece 1 ve 0 değerlerini kullanmak yeterli olmazken, 400×400 boyutunda siyah beyaz bir görüntü dijitalleştirilip renklendirilirken, 400×400 boyutunda bir dizi oluşturulur. Renklendirme işlemi için 0 ve 1 değerlerinden oluşan 2 adet değişken kullanmak yeterli olacaktır.



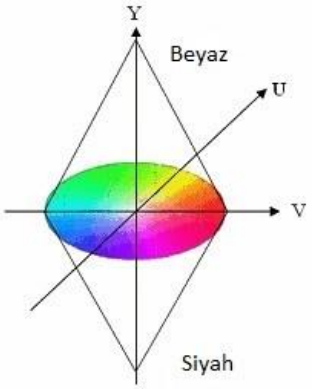
a) RGB Renk uzayı



b) CMYK Renk uzayı



c) HSV Renk uzayı



d) YUV Renk uzayı

Şekil 3.4. RGB, CMYK, HSV, YUV Renk uzayları [36]

Resim dosyaları üzerine veri gizleme işlemi yapılırken birçok farklı yöntem kullanılabilir. Bu yöntemler taşıyıcı nesneye müdahale etme şekline göre temel olarak iki sınıfta incelenebilir. Bu sınıflar:

1. Uzamsal alan (Bit uzayı)
2. Dönüşüm alanı (Frekans uzayı) olarak karşımıza çıkmaktadır [40].

Uzamsal alan yaklaşımlar içersinde en çok tercih edilen ve kullanılan tekniklerin başında LSB (En az önemli bit) tekniği öne çıkmaktadır. Bunun dışında Frekans dönüşümü yaklaşımlarında ise DCT (Discrete Cosine Transforms – Ayrık Kosinus Dönüşümü), DWT (Discrete Wavelet Transforms – Ayrık Dalgacık Dönüşümü), DFT (Discrete Fourier Transforms – Ayrık Fourier Dönüşümü) ve SVD (Singular Value Decomposition – Tekil Değer Ayrışımı) teknikleri gelmektedir [38, 39, 40]. Bu teknikler içinde taşıyıcı nesne olarak genellikle gri-seviye resimler ve renkli sayısal resimler kullanılmaktadır. Renkli sayısal resim dosyaları 24 bit veya 8 bit olarak oluşturulabilir.

24 bit görüntüler: Görüntüyü oluşturan her bir pixel için 3 byte lık değerlerden oluşmaktadır. Üç ana renk bir araya gelerek bir pixel lik görüntü elde edilir. Bu ana renkler Kırmızı (red), Yeşil (green), Mavi (blue). Üç bitlik bir veriyi bir pixel içerisine saklayabiliriz. 24 bitlik 600x400 çözünürlüğüne (pixel sayısı) sahip resim, veri saklamak amacıyla kullanılacak olursa, (600 * 400 * 24)/8 = 720.000 bitlik (yaklaşık 703 byte) veri saklama kapasitesi olacaktır. Gizlenecek veriyi sıkıştırma işlemine tabi tutduktan sonra veri gizleme işlemi yapılacak olursa daha fazla veriyi saklama imkaanı elde ederiz.

RGB renk uzayının nitelemesini temel alarak, “E” verisinin ikili (binary) tabanda karşılığı olan “01000101” bilgisini 3 pixel’e gizleyelim. Orjinal görüntü bitleri şu şekilde olsun:

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

İçine bilgiyi gizlediğimizde oluşan yeni pixel değerleri ise şöyledir:

10010100 00001101 11001000	(148, 13, 200)	→	010
10010110 00001110 11001011	(150, 14, 203)	→	001
10011110 00010001 11001011	(158, 17, 234)	→	01

Gösterilen bu örnekte verileri saklamak için her bir pixel'in sadece son 1 biti kullanılmıştır. Elde edilen sonuçlara bakılacak olursa her üç pixel üzerinde de değişklikler olduğu görülmektedir. Üçüncü pixel'in son sekiz bitini temsil eden mavi (blue) değerinin bitlerini kullanmaya gerek kalmamış diğer sekizli blokların son bitlerinin bazılarında değişiklik olmuştur. Ancak meydana gelen bu değişiklikler insan duyu organlarının algısından uzaktır. Gizlenecek olan bilginin boyutuna ve taşıyıcı nesnenin kapasitesine göre veri saklamada kullanılacak olan bitlerin sayısı artırılabilir. Tabi bu işlem gerçekleştirilirken taşıyıcı nesnedeki bozulmalar gözönüne alınmalıdır. Kapasite miktarı artırılırken algılanamazlık özelliği gözardı edilmemeli.

8 bit görüntüler: Orijinal resme ait pixellerin aşağıdaki gibi olduğunu kabul edersek. Renk paletindeki değerler; beyaz, beyaz, mavi, mavi = 00 00 10 10

10 sayısının ikli tabanda (binary) karşılığı olan 1010 değerini bu pixellere gizlemek istersek, yapılan değişiklikler sonucunda elde edilen yeni pixel değerlerimiz şöyle oluşur:

01 00 11 10 = kırmızı, beyaz, yeşil, mavi

8 bitlik reknli görüntüler ile veri gizleme işlemi yapılacak olursa, bitlerde meydana gelecek olan bir değişim resmin görüntüsünü büyük ölçüde değiştirebilir. Bu sebeple daha önceki çalışmalar da göz önüne alındığında, gri-seviye resimlerin 8 bitlik renkli resimler yerine kullanılması daha doğru bir yaklaşım olacaktır [43].

Görüntü (video) steganografi işlemleride resim steganografisine benzer şekilde oluşturulmaktadır. Bir video, çerçeve (frame) adı verilen resimlerin saniyede ardışık şekilde gösterilmeleriyle oluşturulur. Gürüntü içerisindeki herbir çerçeve (frame)

bilgi saklamak için kullnılmaktadır. Tabi sahip olduğu büyüklükleri göz önüne aldığımızda yüksek bir veri saklama kapasitesi olduğunu söyleyebiliriz.

3.6.3. Ses (Audio) steganografi ve kullanılan teknikler

Ses Steganografisi özellikle 2000'li yılların başında çalışılmaya başlanmış olan, gizli bilgiyi bir ses-video dosyası (wav, mp3, avi, midi, mpeg) üzerine saklamayı hedefleyen veri gizleme şeklidir. Ses steganografisinde dikkat edilmesi gereken iki önemli kıstas vardır. Bunlar [45];

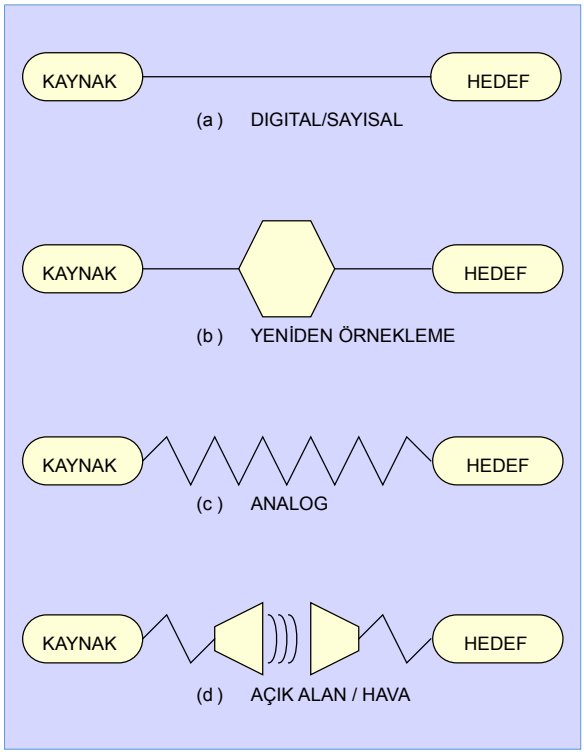
1. Verinin saklanacağı sayısal ses dosyasının türü ve yapısı
2. Gönderici ve alıcı arasındaki iletişim kanalıdır

Ses dosyasının türü ses steganografi yaklaşımlarında büyük önem arzeder. Bilinen ses dosyası türlerinden MP3 dosyaları sıkıştırma işlemine tabi tutulduktan sonra elde edilen ses dosyalarıdır. Bu bakımdan boyutları diğer dosya türlerine göre daha küçüktür. Veri iletiminde bir nesnenin küçük boyutta olması transfer işlemi için oldukça avantajlıdır. Ancak bu tür dosyalara veri gizlemek diğerlerine (sıkıştırılmamış olan dosyalar) oranla daha zor ve sınırlı kapasitede olmaktadır. MP3 dosyalarına veri saklama işlemi ise ses verileri üzerine yapılan sıkıştırma esnasında gerçekleştirilmektedir [44].

Çoğu dijital ses gösterimi için iki kritik parametre vardır; örnek niceleme yöntemi ve geçici örnekleme oranı. Yüksek kaliteli dijital ses örneklerini temsil etmek için en popüler format, 16-bit doğrusal niceleme, Windows Audio-Visual (WAV) ve Ses Değişim Dosyası Formatı (Audio Interchange File Format - AIFF) 'dir. WAV dosyaları ham/işlenmemiş halde oldukları için bu dosyalar üzerinde işlem yapmak ve veri gizlemek daha kolaydır [45, 28, 46, 61]. WAV dosyaları içerisinde 16 bit olarak kodlanmış olanlar ise veri saklama oranı ve az gürültü oluşturması bakımından en uygun olan ses dosyalarıdır.

Yapılan bu çalışmada da WAV dosyaları üzerinde çalışılmıştır. Bu çalışmada olduğu gibi ses steganografisi üzerine yapılan diğer çalışmalarda genelde LSB yöntemi [47, 48] ve dönüştürme teknikleri kullanılmıştır. İnsan duyuları içersinde işitme duyusu görme duyusuna göre daha hassas olduğunu gözönüne alırsak, ses sinyallerine veri gizlemek resim piksellerine oranla daha karmaşık işlemler gerektirir.

Ses iletim ortamları ses için bir veri gizleme yöntemi geliştirilirken dikkate alınması gereken en önemli unsurlardan biridir. Ses iletim ortamı ses sinyalinin gönderici ve alıcı arasında gideceği olası ortamlardır. Bir sinyalin kodlayıcıdan kod çözücüye kadar geçebileceği birçok farklı iletim ortamı vardır. Şekil 3.5.'de bu iletim yöntemlerinden bazıları örnek olarak gösterilmiştir [45]. Şekil 3.5.a'da gösterilen sayısal iletişim kanalı kullanılacak olursa veri gizleme işleminde karşılaşılacak olan kısıtlamalar en aza indirgenebilir.



Şekil 3.5. Ses iletim yöntemleri [45]

Bir sonraki değerlendirme, bir sinyalin daha yüksek veya daha düşük bir örnekleme hızına yeniden örneklenmesidir, ancak tümüyle dijital kalmaktadır (Şekil 3.5.b). Bu dönüşüm, sinyalin çoğunun mutlak büyüklüğünü ve fazını korur, fakat sinyalin zamansal özelliklerini değiştirir [45].

Üçüncü durum, bir sinyalin bir analog duruma "oynatılması", makul şekilde temiz bir analog hat üzerinde iletilmesi ve yeniden örneklenmesini göstermektedir (Şekil 3.5.c). Mutlak sinyal büyüklüğü, örnek niceliği ve zamansal örnekleme oranı korunmaz. Genel olarak, faz korunacaktır [45].

Son durum, sinyalin "havada oynatıldığı" ve "mikrofonla yeniden örneklendirildiği" (Şekil 3.5.d) yapıdır. Sinyal, faz değişimleri, genlik değişimleri, farklı frekans bileşenlerinin kayması, ekolar vb. etkenler ile sonuçlanan, muhtemelen bilinmeyen doğrusal olmayan bozulmalara maruz kalacaktır [45].

Veri gizleme yöntemi seçerken sinyal gösterimi ve iletim yolu dikkate alınmalıdır. Veri hızı, örnekleme oranına ve kodlanan sesin türüne çok bağlıdır. Tipik bir değer 16 bps'dir, ancak bu değer 2 bps ile 128 bps arasında olabilir. Ses steganografi işlemlerinde, verinin saklanacağı dosya türü ve yapısı, iletişimde kullanılacak kanal gözönüne alınarak birden çok farklı yöntemle veri gizleme işlemi yapılmaktadır. Bu bağlamda ses dosyalarına veri gizleme yöntemleri temel olarak dört farklı şekilde gerçekleştirilmektedir [45]. Bu yöntemlerin güçlü ve zayıf yönleri Tablo 3.3.'de gösterilmiştir.

1. Düşük bit kodlaması (Low Bit Encoding/Least Significant Bit-LSB)
2. Faz/ Aşama kodlaması (Phase coding)
3. Tayf yayılması (Spread spectrum)
4. Yankı veri saklaması (Echo data hiding)

Tablo 3.3. Ses steganografisinde kullanılan yöntemlerin karşılaştırılması [49]

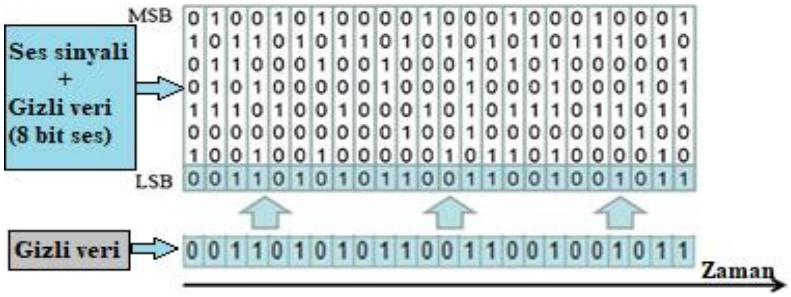
Metod	Veri Gömme Tekniği	Güçlü	Zayıf	Gizleme Oranı
Düşük bit kodlaması	Gizli verinin her biti ses sinyallerinin en az anlamlı bitlerine gömülür	Bilgiyi yüksek hızda gizlemenin etkin ve basit yolu	Gizli bilgiyi ses dosyasından ayıklamak ve yok etmek kolaydır	16 Kbps
Yankı ver gizlemesi	Kaynak sinyale yankı/eko ekleyerek verileri gömer.	Ses dosyası üzerine uygulanabilecek kayıplı sıkıştırma algoritmalarına karşı dayanıklıdır.	Düşük güvenlik ve düşük gizleme kapasitesi	40-50 Bps
Aşama/Faz kodlaması	Kaynak sinyali parçalara böler ve faz referanslarını değiştirir.	Sinyal algılanan gürültü oranı (SPNR) açısından duyulmayan bir kodlamaya ulaşır.	Düşük veri gizleme kapasitesi vardır.	333 Bps
Eşlik Kodlaması	Orijinal sinyali ayrı örneklere ayırır ve gizli mesajın her bir bitini bir eşlik biti içerisine katıştırır	Gönderenin gizli biti kodlamada daha fazla seçeneği vardır.	Sağlam değildir.	320 Bps
Tayf yayılması	Gizlenecek verileri bütün ses sinyalleri üzerine yayar.	Sağlamlık daha iyi sağlanır	Zaman ölçekli modifikasyona karşı savunmasızdır.	20 Bps

Verilen bu dört tekniğe ek olarak eşlik kodlaması (parity coding) tekniğide ilave edilebilir. Eşlik kodlama tekniği temelde LSB yöntemini baz alarak geliştirilmiş olan bir yöntem olarak karşımıza çıkmaktadır. Tablo 3.3.'de bu veri gizleme yöntemlerinin kullandıkları teknik, gizleme kapasitesi, güçlü ve zayıf yönlerinini gösteren bilgiler verilmiştir. Bu verilerden de anlaşılacağı üzere ses dosyası üzerinde veri gizleme kapasitesi bakımından en etkin yöntem LSB yönteminin olduğu görülmektedir.

Bu tez çalışmasında yapılan uygulamada gizli verileri gömmek için taşıyıcı ortam olarak ham ses dosyası (wav) kullanılmış, bu ses dosyası içerisine LSB tekniği kullanılarak veri gizlemesi yapılmıştır. Bu sebeple ses dosyaları üzerine veri gizleme tekniklerine biraz daha detaylı bakmak faydalı olacaktır.

3.6.3.1. **Düşük bit kodlaması (Least significant bit-LSB)**

LSB (Least signaficant bit - En az anlamlı bit) olarak da bilinen bu yöntem, veri gizlemek için kullanılan ilk yöntemlerden biridir [45]. Resim dosyalarında olduğu gibi ve Şekil 3.6.'da gösterildiği gibi, gizli verideki her bir bit, örtü/taşıyıcı sesinin en az anlamlı bitlerine belirleyici bir şekilde gömülmesine dayanır. Böylece 16 kHz örneklenmiş bir ses için 16 kbps'lik veri gizlenmiş olacaktır. Ayrıca resim dosyalarında olduğu gibi ses dosyalarında da birden fazla bit'e veri saklanabilir. LSB yöntemi, taşıyıcı ses dosyasında gizlenecek veri için yüksek gömme kapasitesine izin verir ve diğer gizleme teknikleriyle uygulanması veya birleştirilmesi nispeten daha kolaydır.



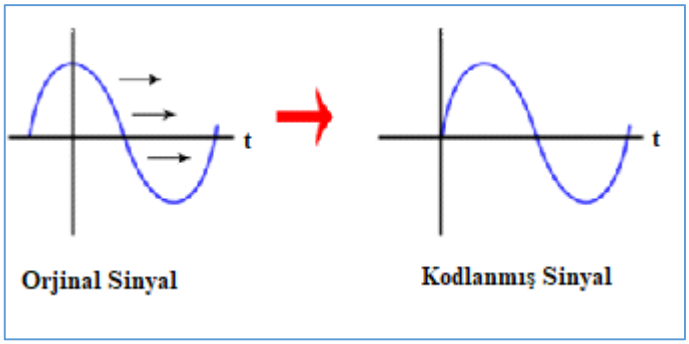
Şekil 3.6.Ses dosyalarında LSB tekniği [49]

Bununla birlikte, bu teknik ile yapılan gizleme işlemlerinde gürültü oranı oldukça düşük olmasına rağmen, yapılacak olan saldırılara karşı dayanıksızdır [45]. Yapılan çalışmalardan da anlaşıldığı üzere ses dosyası üzerine yapılabilecek olan filtreleme, sıkıştırma, sinyali genişletme/yükseltme, gürültü ekleme gibi saldırılar karşısında gizli verinin yok edilmesi veya bozulması muhtemeldir. Ayrıca, veriler çok belirleyici bir şekilde gömülü olduğundan, bir saldırgan tüm LSB düzlemini kaldırarak mesajı kolayca ortaya çıkarabilir [45, 49]. Bu bakımdan steganaliz ataklarına karşı gizli bilgi güvenliğini sağlayabilmek için veri gömme işleminde kullanılacak algoritma büyük önem taşımaktadır.

3.6.3.2. **Aşama/Faz kodlaması (Phase coding)**

Aşama/Faz kodlama tekniği kullanılarak yapılan steganografik yaklaşımda, segmentlere bölünmüş olan ses dosyasının her bir segmentine ait faz değeri saklanacak veriye ait olan faz referansı ile değiştirilmesi yolu izlenir. Bu kodlama tekniğinde gerçekleştirilen işlemler insan duyu organlarının algısından uzak olan sesler üstünde çalışılmaktadır [45, 49, 50]. Faz kodlaması, kullanılabileceği zaman, sinyal-üst düzeyli gürültü oranı açısından en etkili kodlama yöntemlerinden biridir. Her bir frekans bileşeni arasındaki faz ilişkisi önemli ölçüde değiştiğinde, fark edilebilir bir faz dağılımı meydana gelecektir. Ancak, fazın modifikasyonu yeterince küçük olduğu sürece fark edilmeyen bir kodlama elde edilebilir. Duyulmazlığı/algılanamazlığı sağlamak için, faz bileşenleri modifikasyonu küçük tutulmalıdır [45, 49]. Aşama kodlaması işleminde veri gizleme adımları aşağıdaki gibidir [45]:

1. Orijinal ses dosyası M adet kadar alt parçaya/segmente ayrıştırılır.
2. Aşama ve büyüklük matrisleri oluşturmak için her bir segment DFT işlemine tabi tutulur.
3. Birbiri ardı sıra gelen alt parçalar arasındaki faz farklılıkları hesaplanır.
4. Gizlinecek verinin eklenmesiyle her bir alt parçanın faz değeri yeniden oluşturulur.
5. Yeni segmentleri oluşturmak için; büyüklük matrisleri ile oluşturulmuş olan yeni faz matrisleri birleştirilir.
6. Şekil 3.7.'de gösterildiği gibi kodlanmış çıkış yeni segmentler birleştirilerek elde edilir.



Şekil 3.7. Aşama/Faz kodlaması [50]

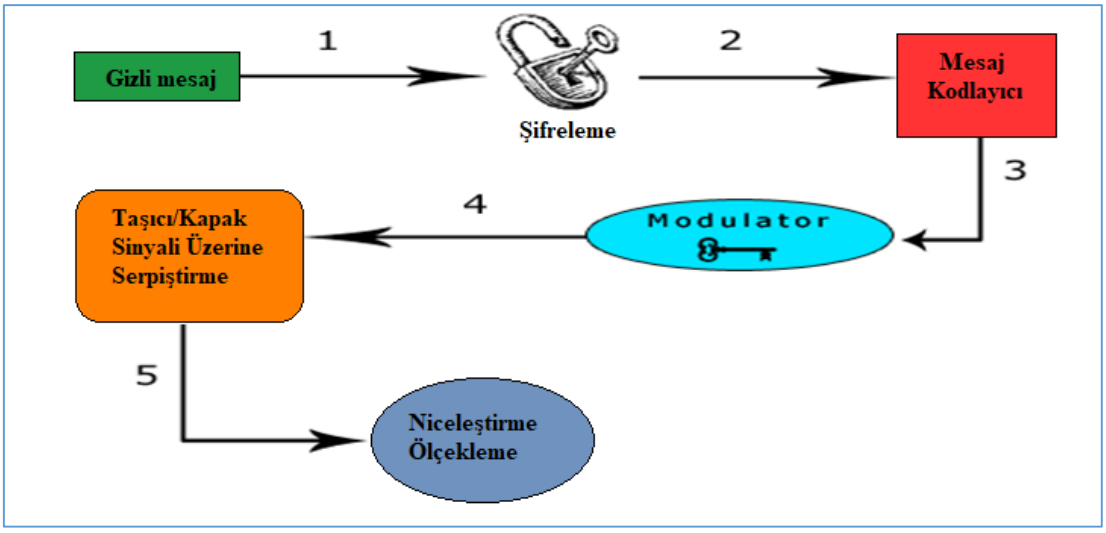
Alıcı taraf gizli veriyi çıkarmak için kendisine ulaşan segmentler üzerinde Ayrık Fourier Dönüşüm (DFT) tekniğini uygular ve dosya üzerindeki faz yeniden elde edilir. Bunu yapabilmesi için de alt parçalara ait toplam segment sayısını bilmesi gerekir [45].

Aşama kodlaması genellikle küçük veri gizleme işlemlerinde kullanılmaktadır. Bunun sebebi taşıyıcı dosya üzerindeki segment sayısına bağlı olarak değişen düşük oranda veri iletim kapasitesine sahip olmasından kaynaklanır.

3.6.3.3. **Tayf yayılması (Spread spectrum)**

Tayf yayılması (Spread spectrum - SS) tekniğinde gönderilmek istenilen gizli mesaj, ses sinyali içerisinde birden çok frekans bandına yayılır. Tayf yayılması tekniği LSB tekniğin analog sesler üzerine uygulanmış halidir [45]. Yüksek oranda veri saklama imkânı vardır. İki yaklaşım arasındaki fark LSB sayısal bitler üzerinde işlem yaparken, tafy yayılması tekniği gerçek ses sinyalinden bağımsız olarak ses verisinin frekans spektrumları üzerine yerleştirir. Tayf yayılması yöntemi genellikle askeri haberleşme içerisinde kullanılan veri gizleme tekniği olarak karşımıza çıkmaktadır

Tayf yayılmasının genel işleyişi Şekil 3.7.'de gösterilmiştir. Şekilde verilen işlem adımlarını açıklayacak olursak:



Şekil 3.8. Tayf yayılması genel işleyiş adımları [50]

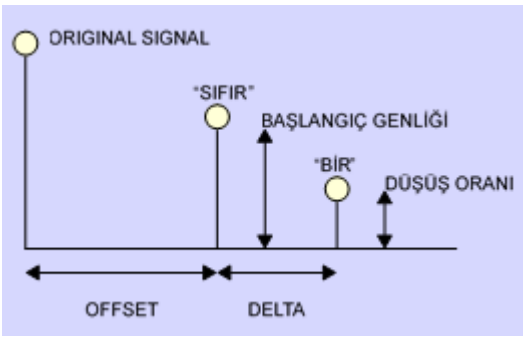
1. Gizli mesaj simetrik anahtar kullanılarak şifrelenir.
2. Şifrelenmiş mesaj, düşük oranlı bir hata düzeltme kodu kullanılarak kodlanır. Bu adım, sistemin genel sağlamlığını artırır.
3. Bu adımda kodlanan mesaj ayrı bir gizli anahtar kullanmak suretiyle, yeni bir sinyal modu üretilir.
4. Gizli mesajı içinde barındıran sinyal, kapak sinyali ile birleştirilir.
5. Son olarak mesaj verisinide içinde barındıran ses dosyası oluşturulur.
6. Mesajı çıkarma işlemleri için bu işlem adımları tersten işletilir [49].

Ses steganografisinde tayf yayılması (spektrum yayma) işlemi ağırlıklı olarak iki tekniğe göre yapılmaktadır. Bu teknikler [51];

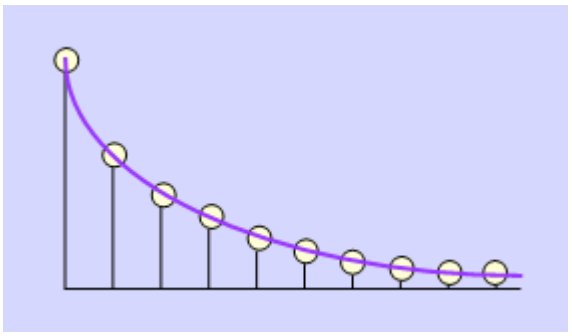
1. Frekans atlamalı yayılmış spektrum (Frequency hopping spread spectrum - FHSS) : FHSS metodunda ses verisine ait frekans spektrumu değiştirilir ve mesaj frekanslar arasında sekerek gizlenmiş olur.
2. Doğrudan sıralı spektrum yayma (Direct sequence spread spectrum - DSSS): Taşıyıcı ses sinyalleri üzerine pseudorandum (rastgele üretilen sinyal) sinyal tarafından ayarlaması yapılan ve içinde gizli mesajı barındıran yeni sinyal ilave edilmek suretiyle işlem gerçekleştirilir.

3.6.3.4. **Yankı veri saklaması (Echo data hiding)**

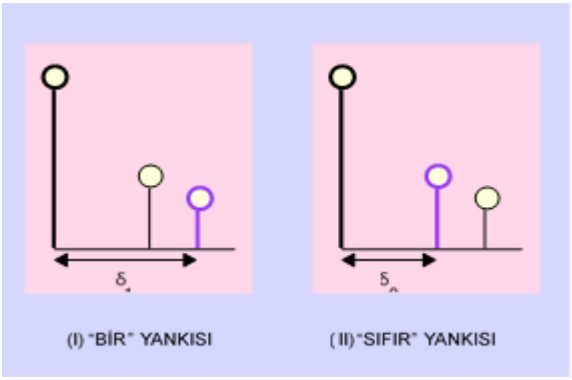
Yankı veri gizleme tekniğinde, gizli veriler yankı olarak isimlendirilen ayrı bir sinyal içerisinde ses dosyasının içine yerleştirilir. Tayf yayılmasında olduğu gibi, yankı veri kodlamasında da yüksek veri iletim oranına imkânı vardır. Şekil 3.9.'da gösterildiği gibi veriyi saklamak için yankının, orijinal sinyalden sapma değeri (gecikme süresi), düşüş oranı, genlik değerleri değiştirilir. Yankı veri kodlamasının algılanamazlık (zor deşifre) ilkesini yerine getiren etkili bir teknik olduğu söylenebilir. Şekil 3.9.'da gösterildiği gibi bir gecikme süresi değeri ikili (binary) 1 değerini temsil ederken, ikinci bir gecikme süresi ikili (binary) 0 değerini temsil eder [45].



a) Ayarlanabilir parametreler



b) Ayrık zaman katsayıları



c) Elde edilen yankı verileri

Şekil 3.9. Yankı veri kodlaması [45]

3.7. Steganografik Yazılımlar

Tablo 3.4. Steganografik yazılımlar [64]

Program	Kullanılan Taşıyıcı Nesne	Açıklama
BMPSecrets	**R:** GIF, TIFF, JPG, BMP	
DarkCryptTC	**R:** GIF, TIFF, JPG, PNG, SD, TGA, TNG **S:** WAV **Doc**: ODT, HTML, XML, TXT **D:** EXE, DLL, NTFS katarları	RSD modu (RNG tabanlı rasgele veri dağılımı) kullanılmıştır. AES şifreleme desteklenir.
DeepSound	**R:** BMP **S:** WAV, MP3, Audio CD, APE tag, FLAC, WMA	256 bit AES şifrelemeyi kullanır
ImageSpyer G2	**R:** TIFF, BMP	RSD (Reciprocal smallest distance algorithm - Karşılıklı en küçük mesafe) algoritmasını kullanır
MP3Stego	**S:** MP3	Açık kaynak
Mr. Crypto	**R:** TIFF, PNG, BMP	AES, 3DES şifrelem algoritmalarını kullanır. LSB tekniğine göre gizleme yapar
OpenPuff	**R:** TGA, PNG, JPEG, BMP **S:** WAV, MP3 **V:** MPEG-2, MPEG-1, MP4, 3gp, VOB, FLV, SWF,	Birden fazla taşıyıcı dosyaya veri gizleyebilir. Kriptografi, steganografi ve beyazlatma tekniklerini içinde barıdırır.
OpenStego	**R:** BMP, PNG	Açık kaynak
Outguess	**R:** JPG	
QuickStego / QuickCrypto	**R:** BMP, JPEG, GIF	
S-Tools	**R:** BMP, GIF **S:** WAV **D:** Kullanılmayan disk alanları	
Steg	**R:** BMP, PNG, JPEG, GIF	Simetrik ve asimetrik şifreleme kullanır. Win/Linux/Mac de çalışır.
StegaMail	**R:** BMP, PNG	56 bit şifreleme yapar. zLib sıkıştırma kullanır.
Steganographic Laboratory (VSL)	**R:** TIFF, PNG, JPG, BMP	Açık kaynak
Steganography Studio	**R:** GIF, BMP, PNG	Açık kaynak Farklı gizleme tekniklerini (LSB, LSB Matching, SLSB) kullanılır.
StegHide	**R:** JPG, BMP **S:** WAV, AU	Açık kaynak
StegoShare	**R:** BMP, JPEG, PNG, GIF, TIFF	Açık kaynak

Önceki konu başlıklarında anlatıldığı gibi steganografik sistemlerin oluşturması için temel gereksinmlerin başında verinin gizleneceği taşıyıcı nesne (cover object) gelmektedir. Günümüzde de bu ortamlar üzerinde veri gizleme işlemini yapan birçok yazılım vardır. Geliştirilmiş olan bu veri gizleme ve maskeleme yazılımları bünyelerinde birden fazla tekniği barındırabilmektedir. Veri gömme işlemi öncesinde, gizlenecek veriler üzerinde şifreleme, sıkıştırma teknikleri birlikte

kullanılabilmektedir. Bu yazılımlardan en çok bilinen ve kullanılanlar Tablo 3.4.'de gösterilmiştir. Verilen bu yazılımlar arasında ses steganografisi üzerine işlem yapan bazı uygulamalar kalın ve yatık şekilde gösterilmiştir. Ayrıca burada verilen bazı uygulamar hakkında daha detalı bilgilere diğer çalışmalarda [61] yer verilmiştir.

3.8. Steganaliz (Sır açma)

Steganaliz, bir taşıyıcı nesne (resim, ses, video, metin) içerisinde, gömülü başka herhangi bir veri barındırıp barındırmadığını tespit eden, devamında tepit edilen gizlenmiş verileri ayıklamaya dayanan işlemleri bünyesinde barındıran testler olarak tanımlanabilir. Sıraçmada dosya içerisinde gizli bilgi varlığı sezilmeye çalışılır. Bu işleme sezme (detection) işlemi denir. Bundan sonra ise gizli bilgiyi çıkarma (extraction) işlemi gelir [46]. Bunların yanı sıra steganaliz işlemleri sadece gizli veri olup olmadığına veya bu verinin taşıyıcı nesneden çıkarılma işlemleri ile ilgi değildir. Aynı zamanda steganografik sistemin algılanamazlık, sağlamlık ve kapasite ölçütlerine göre güçlü veya zayıf yönlerini ortaya koymak için yapılabilir.

Taşıyıcı/örten ortamların çeşitliliği ve bu nesneler üzerine veri gizlemede kullanılan tekniklerin farklılıkları gözönüne alındığında, her stego sistem üzerinde istenilen doğru sonucu elde edebilecek bir analiz yaklaşımı henüz ortaya konmamıştır. Her steganografik yapı farklı bir steganaliz çalıması gerektirebilir. Ancak çoğu steganaliz uygulamasının temeli matematiksel ve/veya istatistiksel analizlere dayanır.

Amaçlarına göre aktif saldırı ve pasif saldırı olmak üzere iki farklı steganaliz yaklaşımı vardır [5];

1. Pasif saldırı: Sadece gizli verinin varlığını tespit eden yöntemler
2. Aktif saldırı: Tespit edilen gizli veriyi geri elde etmeyi amaçlayan yöntemler.

Steganaliz yaklaşımlar, taşıyıcı/örten ortam olarak kullanılan kapak nesnesi üzerinde verilerin gizlendiği alanlarda işlem yaparlar. Steganaliz yaklaşımları işlem yaptığı düzleme göre üç kategori olarak sınıflandırılmıştır. Bu kategoriler;

1. Uzamsal düzlem üzerinde işlem yapan yaklaşımlar (Resim)
2. Zaman düzlemi üzerinde işlem yapan yaklaşımlar (Ses)
3. Hem uzamsal hem de zaman düzlemi üzerinde çalışan yaklaşımlar (Video)

Bir steganografik yapı incelenirken steganografinin üç temel özelliği dikkate alınır. Bu özellikler daha önceki konu başlıklarında da aktarıldığı gibi şunlardır [35, 51];

1. Algılanamazlık / Taşıyıcı nesnde meydana gelen değişim
2. Dayanıklılık / Sağlamlık
3. Kapasite

Olarak nitelendirilmektedir. Steganografik sistemler içerisinde bu özelliklerin kendi aralrında bir ikilemi olduğunuda tekrar hatırlatmakta fayda olacaktır. Karşılan bu ikilemler veya ödünleşimler şunlardır [51];

1. Dayanıklılık ~ Kapasite
2. Algılanamazlık (Değişim) ~ Kapasite

İkilemleri / ödünleşimleri yaşanmaktadır.

3.8.1. Steganalizde saldırı türleri

Steganalist; içerisine bilgi gizlenmiş steganografik yapılara çeşitli ataklar düzenleyerek gizli bilginin varlığını ve ortaya çıkarılmasını hedefleyen kişidir. Saldırı yapılacak olan steganografik yapı bilinmiyorsa bu durum steganalist için büyük bir dezavantaj oluşturur. Bu sebeple işlem yapılacak stego yapılar hakkında bazı bilgilerin önceden bilinmesi veya öngürülmesi gerekir. Steganalist sahip olduğu eldeki verilere göre Tablo 4.5.'de verilen saldırı yöntemlerinden birini seçebilir. Bölüm 2.4.7.'de anlatıldığı gibi steganalizde kullanılan bu yaklaşımlar kriptanalizde kullanılan saldırılara benzerlik göstermektedir. Steganaliz ataklarında bir steganalist elindeki bilgiler veya varsayımlar doğrultusunda çalışmalarını gerçekleştirir. Bu ataklar taşıyıcı/kapak ve gizli mesajın yerleştirildiği stego nesne, bilinen taşıyıcı,

bilinen mesaj, seçilen stego, seçilen mesaj, bilinen stego ve evrensel tespit yöntemleridir [5,55].

Tablo 3.5. Steganaliz saldırı çeşitleri [5, 46, 55]

Saldırı türü	Açıklama
Sadece stego saldırısı (stego-only)	Kriptanalizde kullanılan sadece şifreli metin atağına benzemektedir. Analiz için sadece stego nesne (Stego-object) vardır.
Bilinen taşıyıcı saldırısı (known cover)	Taşıyıcı nesne ile sırlı/stego nesnenin ikisinin de bilindiği saldırı türüdür.
Bilinen mesaj saldırısı (known message)	Stego nesneyi analiz etmek için gizlenmiş verinin bilinmesi temeline dayanır. Sadece stego (stego-only) saldırısında olduğu gibi bu saldırı şeklide oldukça zordur.
Seçilen stego saldırısı (chosen stego)	Stego nesne ve uygulanan steganografik tekniğinin bilindiği durumlardaki uygulanan saldırı yöntemidir.
Seçilen mesaj saldırısı (chosen message)	Bu saldırı türünde steganalist çeşitli steganografi algoritmalarını eldeki bilinen mesajlar üzerine kullanarak stego nesneler üretip, bilinen stego nesneyi elde etmeyi hedefler. Böylece stego yapı içerisinde kullanılan modeli bulmayı amaçlar.
Evrensel tespit	Evrensel tespit çalışmalarında saklama algoritmasına ihtiyaç duyulmaz ve bu tesbit her türlü saklama algoritması için geçerlidir. Bu şekildeki çalışmalar daha çok önem kazanmıştır [46].

Bu saldırılar sonucunda temel olarak hedeflenen; var ise gizli verinin tespit edilebilmesini sağlamak, varlığı tespit edilen mesajı yeniden elde ederek gizleme için kullanılan tekniği/algoritmayı ortaya çıkarmak veya bu gizli mesajı yok edebilmektir.

3.8.2. Steganalizde kullanılan ölçüm yöntemleri

Bir steganografik algoritmanın başarısının değerlendirilmesinde bakılan en önemli unsur taşıyıcıda (cover object) ne kadar değişim meydana getirdiğidir. Bununla birlikte steganaliz ataklarına karşı göstereceği dayanıklılık ayrı bir ölçülm kriteridir. Steganaliz taşıyıcı nesnedeki değişiklikleri analiz ederek dosya içerisinde gizli verinin varlığını bulmaya çalışır. Gizli verinin tespitini zorlaştırmak için taşıyıcı dosyada üzerinde meydana gelebilecek değişim oranını minimum seviyede tutmak gerekir. Taşıyıcıdaki değişimi ya da bozulma oranının belirlenmesi için çeşitli ölçme yöntemleri vardır. Bu yöntemler arasında en çok bilinen ve kullanılan yöntemler şunlardır [51,55,56,59]:

1. Ortalama mutlak hata (Mean Absolute Error - MAE)
2. Ortalama karesel hata (Mean Squared Error - MSE),
3. Ortalama karekök hata (Root Mean Squared Error - RMSE),
4. Sinyal gürültü oranı (Signal to Noise Ratio - SNR)
5. Tepe/Doruk sinyal gürültü oranı (Peak Signal to Noise Ratio - PSNR)'dır.

Yukarıda verilen bu yöntemler aşağıda hesalama formülleri ile beraber kısaca anlatılmıştır. Denkelemlerde kullanılmış olan; N toplam sinyal/işaret sayısını, i sinyal dizisinin ilgili elemanını, X orijinal taşıyıcı nesneyi, Y ise stego nesneyi ifade etmektedir.

Ortalama Mutlak Hata (MAE): MAE, orjinal ve stego nesnelere ait vektörlerin yönlerini dikkate almadan bir dizi tahmindeki hataların ortalama büyüklüğünü ölçer. Sürekli değişkenler için doğruluğu ölçer. MAE değerinin hesaplanmasında ölçüm yapılan orijinal ve stego nesnelere ait her bir dizi/vektör arasındaki farklılıklar ortalamada eşit olarak ağırlıklandırılıt ve doğrusal bir sonuç elde edilir. MAE hesaplama formülü Denklem 3.1.'de verilmiştir.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} | X_i - Y_i | \qquad (3.1)$$

Ortalama Karesel Hata (MSE): MSE hataların (orijinal nesne – stego nesne) kareleri toplamının ortalamasıdır. MSE, genellikle sinyallerde iki sinyalin birbirilerine olan benzerliklerini ölçmek için kullanılan bir yöntemdir ve σ2 olarak gösterilir. MSE değerinin 0 (sıfır)'a yakın olması başarılı bir gizlemenin yapıldığı anlamına gelir. Hesaplama formülü Denklem 3.2.'de gösterildiği gibidir [40,51,57].

$$MSE = \sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (X_i - Y_i)^2 \qquad (3.2)$$

Ortalama Karekök Hata (RMSE): RMSE değeri ise MSE'nin karekökü olarak hesaplanır. RMSE hesaplama formülü Denklem 3.3.'de gösterilmiştir.

$$RMSE = \sqrt{MSE} = \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - Y_i)^2} \quad (3.3)$$

Sinyal Gürültü Oranı (SNR): SNR, taşıyıcı/orjinal ses dosyası sinyallerinin, stego ses sinyallerindeki gürültüye (bozulmaya) olan oranıdır. Veri gömme işleminden sonra ses dosyasındaki çıkış sinyalinin kalitesinin bir ölçüsüdür. Desibel (dB) cinsinden ölçülür. Yapılan hesaplama sonucunda 30 dB veya daha büyük bir SNR değerinin elde edilmesi ses dosyası kalitesinin korunduğu manasına gelmektedir [49, 58, 61]. SNR hesaplanması formulü Denklem 3.4.'de verilmiştir.

$$SNR_{dB} = 10 * \log_{10}\left(\frac{\sum_{i=1}^{N}(Xi)^2}{\sum_{i=1}^{N}(Xi - Yi)^2}\right) \quad (3.4)$$

Tepe/Doruk sinyal gürültü oranı (PSNR): PSNR filigranlama veya ses steganografi tekniklerinin uygulandığı bir stego sistemin ortaya koyduğu algılanamazlık özelliğinin performansının ölçülmesinde kullanılmaktadır. Ölçü birimi Decibel (dB) dir. Sesin kalitesinin korunup korunmadığı PSNR değerinin yüksek olup olmaması ile doğru orantılıdır. PSNR hesaplanırken MSE değerinden yararlanılır [40, 51, 57, 61]. PSNR hesaplama formülü Denklem 3.5.'de gösterilmiştir.

$$PSNR(dB) = 10\log_{10}\frac{X_{peak}^2}{MSE} \quad (3.5)$$

Bu denkelemde X_{peak}^2 olarak gösterilmiş olan değer; ses dosyası içerisndeki en büyük değere sahip olan sinyalin karesidir.

3.8.3. Steganaliz teknikleri ve mevcut steganaliz yazılımları

Steganaliz işlemlerinde öncelikle bir dosya içerisinde gizli bilgi varlığı sezilmeye çalışılır. Bu işleme sezme (detection) saldırısı denir. Bu saldırılar stego nesne içerisinde gizli veri olup olmadığını algılamaya çalışırlar. Sıraçma işlemlerinde sezme saldırısı olarak kullanılan çeşitli yöntemler mevcuttur. Bu yöntemlerden en yaygın olarak bilinen ve kullanılanlar şunlardır [46, 60, 74]:

- $\chi 2$ (Ki-Kare) Testi
- Histogram Analizi (PoVs'lerin Analizi)
- RS steganaliz (İkili istatistik yöntemi)
- RQP yöntemi (Raw Quick Pairs)
- JPEG dosyalarda steganaliz
- Görsel/İşitsel ataklar
- Evrensel tespit yöntemleri

Bu sezme saldırıları daha önce yapılan diğer çalışmalarda [28,46,55,60,74] detaylı olarak anlatılmıştır. Bu çalışma içerisinde yapılan steganografi uygulamasının analizinde kullanılan $\chi 2$ (ki-kare) testi bir alt başlıkta anlatılmıştır.

Yukarıda verilen bu yöntemleri kullanarak geliştirilmiş olan steganaliz yazılımları mevcuttur. Bu yazılımlar steganografik yaklaşımları gözönüne alarak geliştirilmiş kısmi çözümler sunan uygulamalardır. Daha önceki konu başlıklarında değinildiği üzere her steganografik yapı kullandığı tekniğe bağlı olarak ayrı bir analiz gerektirebilir. Geliştirilen bu yazılımlar bazı tekniklere göre hazırlanmış olan gizli veri çıkarımı yapabilmektedir. Bu yazılımlarda öncelikli olarak gizli verinin çıkarılması değil, gizli verinin tespit edilebilmesi hedeflenmiştir. En çok bilinen ve kullanılan bazı steganaliz yazılımları şunlardır [55, 60, 61, 74]:

- StegSpy
- Stegdetect
- StegBreak

- Stego Suit
- Stegoanalyzer

Bu yazılımlarda StegSpy ve Stegdetect programları açık kaynak kodlu uygulamalar olup sadece belirli programların üretmiş oldukları stego resimleri tespit edebilmektedir. StegSpy JP Hide and Seek, Invisible Secrets, Hiderman, JPegX, Masker steganografi programlarını, Stegdetect ise Jsteg, jpHide, InvisibleSecrets, Outguess, F5, Camouflage adlı steganografi programlarını tespit edebilmektedir. StegBreak yazılımı ise steganografinin tespit edilmesine yönelik bir program değildir. Daha çok steganografi uygulanmış ise içerisindeki mesajın çıkartılmasına yönelik bir uygulamadır. Hedef aldığı steganografi programları ise JstegShell, JPHide ve Outguess'dir [55].

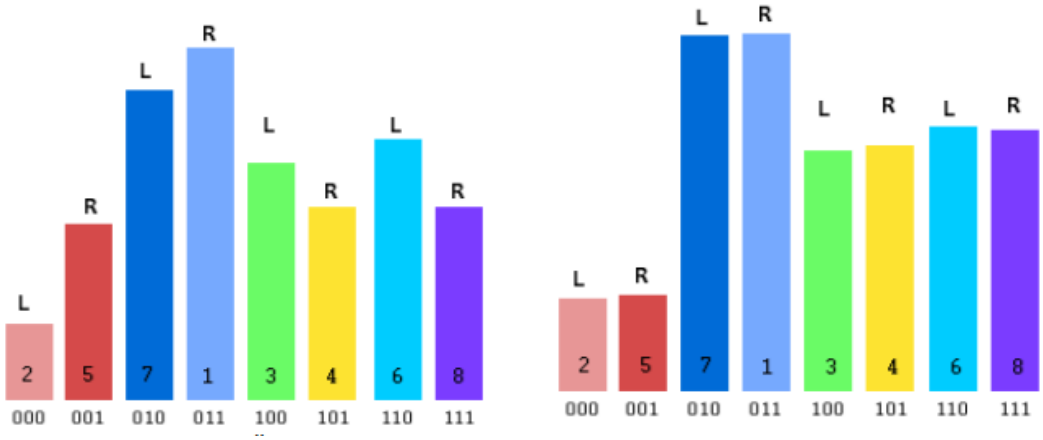
Bu yazılımlar ile ilgili olarak daha detaylı bilgiler daha önce yapılmış olan çalışmalarda mevcuttur [60, 55, 61].

3.8.3.1. **χ2 (ki-kare) testi**

Steganografik yapılar üzerinde birçok farklı teknik kullanmak suretiyle analizler yapılmış ve bu analizler sonucunda ilgili stego nesne üzerinde gizli verinin varlığı tespit edilmeye çalışılmıştır. Uygulanan steganografi tekniğine bağlı olarak gerçekleştirilen bu steganaliz testlerinin başında χ2 (ki-kare) testi gelmektedir. χ2 testi özellikle LSB tekniği uygulanarak gerçekleştirilen stego yapılar üzerinde oldukça doğru sonuçlar veren istatistiksel bir analiz yöntemidir.

χ2 (ki-kare) testi gözlenen ve beklenen frekanslar arasındaki (PoVs Pair of Values-Değer Çifti) farkın anlamlı olup olmadığı temeline dayanır ve çoğunlukla niteliksel olarak belirtilen verilen analizinde kullanılır. Ayrık Kosinüs Dönüşümü (DCT) tekniğini üzerine kurulu olan ki-kare testi Westfeld ve Pfitzmann tarafından 2000'li yıllarda geliştirilmiş ve ilk olarak görsel/resim dosyaları üzerinde kullanılmıştır [75]. LSB bitleri üzerinde veri gizleme yapılan nesnelerde yapılan analizlere incelendiğinde, değer çiftleri (PoVs-DÇ) frekansları düz/düzgün bir dağılım

göstermektedir. Yani tekil değer çiftleri ile çift değer çiftlerinin arasındaki frekans farkları azalmış birbirine eşit hale gelmiştir. Normalde bu değer çiftlerinin frekansları düz bir şekilde dağılmamaktadır. DCT katsayıları ve elde edilen indislerin freakans dağılımları Şekil 3.10.'da gösterilmiştir.



a) Orijinal nesne b) Stego nesne

Şekil 3.10. Resim içerisine veri gizlenmeden önce ve sonraki frekans dağılımları

Geliştirilen ki-kare testi ile sayısal resimler ve videolar üzerinde gerçekleştirilen analizler sonucunda değerler "0" düzeyinde çıkmaktadır. Bu analizler düşük çözünürlüklü sayısal ortamlarda daha iyi sonuç vermektedir. Ki-kare testi rastgele karmaşık düzende veri gizlenmiş bir nesne üzerinde iyi bir sonuç vermemekle beraber, sıralı ve düzenli bir saklama yapılan stego nesneler üzerinde yapılan analizler oldukça etkin ve sağlıklı sonuçlar vermektedir ve "0" dan farklı değerler elde edilmektedir. Ayrıca gizlenen verinin boyutunun/oranın da bulanabilmesi bu test ile mümkündür. LSB bitleri üzerine veri gizlenmiş nesneler üzerinde etkin sonuçlar veren ki-kare analizinin algoritma adımları aşağıda verilmiştir.

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

Adım 2. Sıralı bir şekilde gizlenmiş mesajın nesne içerisine gizlenmesinden sonra, i kategoride beklenen frekans dağılımı aşağıdaki gibidir.

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

Adım 3. Rastgele örnekleme sonucu elde edilen frekans değeri ise aşağıdaki gibidir.

$$n_i = |\{renk\}|(renk'in\ sıralı\ indeksi \in \{2i, 2i+1\})|$$

Adım 4. k-1 bağımsızlık dereceleri ile elde edilen $\chi 2$ istatistik değeri şu şekilde hesaplanır.

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

Adım 5. $n_i\ ve\ n_i^*$ Dağılımlarının eşit olduğu durumlarda, gizli veri gömme olasılığı *P* değeri elde edilir. Bu olasılık yoğunluk fonksiyonun integrali alınarak hesaplanmaktadır. (Verilen hesaplama formülünde kullanılan Γ, Euler'in ortaya koyduğu gama fonksiyonudur.) Gama fonksiyonu matematikte faktoriyel fonksiyonun karmaşık sayılar ve tam sayı olmayan reel sayılar için genellemesi olan bir fonksiyondur [76].

$$P = 1 - \frac{1}{2^{\frac{k-1}{2}}\,\Gamma\left(\frac{k-1}{2}\right)} \int_0^{x_{k-1}^2} e^{\frac{x^2}{2}} \cdot x^{\frac{k-1}{2}-1}\, dx$$

Provos, LSB tekniği uygulanarak yapılan steganografi uygulamaları üzerinde oldukça etkili bir sonuç veren $\chi 2$ testini, test aralıklarını ve değerleri yeniden değiştirmek suretiyle genişletmiştir. Bu genişletme işlemini yaparken P ve (P+1) değer çiftlerinden P ve (P-1) çifte kadar yeniden hesaplama yoluna gitmiştir [76].

BÖLÜM 4. MATERYAL VE GELİŞTİRİLEN UYGULAMA

Kriptografi (veri şifreleme), steganografi (veri gizleme) ve veri sıkıştırma tekniklerini beraber kullanılarak hazırlanmış olan bu çalışma C# programlama dili kullanılarak Visual Studio. NET ortamında gerçekleştirilmiştir. Gerçekleştirilen bu stego yapının sağlamlığı ve başarısını test etmek için; steganaliz için kullanılan SNR, PSNR değerlerinin ölçümleri Matlab ortamında yapılmıştır. Ayrıca oluşturulan stego yapıların dayanıklılık ve algılanamazlık ölçümleri için $\chi 2$ (ki-kare) testi, Histogram analizi, spectrogram analazi uygulanmış ve bu testlerinin sonuçlarına da bir sonraki bölüm altında yer verilmiştir.

Uygulama içersinde sözlük oluşturmak için içerisinde Türkçe karakterlerin de yer aldığı genişletilmiş ASCII karakter tablosundan faydalanılmıştır.

Gizlenmek istenilen verilerinin sıkıştırılma işlemi için kayıpsız veri sıkıştırma algoritmalrından olan LZW (Lempel Ziv Welch) tekniği kullanılmıştır. Ses dosyası içerisine gizlenmek istenilen veriler, temeli LZW algoritmasına dayanan ve bu algoritmalardan türetilmiş olan LZW, LZM, LZMA, 7ZIP, DEFLATE algoritmalardan biri ile sıkıştırılmıştır. Bu işlem ile ses dosyası içerisine gizlenmek istenilen verilerin boyutları, verinin uzunluğuna ve türüne göre farklı oranlarda küçültülmüştür. Gizli/Saklanmak istenilen veriler üzerinde uygulanan sıkıştırma teknikleri sonucunda, elde edilen sıkıştırma oranlarının başarısı sonraki bölümde kapasite analizi konu başlığı altında ayrıca verilmiştir.

Geliştirilen uygulamada veri sıkıştırma işleminin yanı sıra güvenliği artırmak amacıyla veri şifreleme seçeneğide sunulmuştur. Bu amaçla verileri şifrelemek için uygulama içerisinde kullanıcının insiyatifinde olmak üzere AES, DES, 3DES simetrik blok şifreleme ve RC4 simetrik dizi (akış) şifreleme algoritma seçenekleri sunulmuştur. Mevcut şifreleme algoritmalarının bilinirliliği gözönüne alınarak bu

şifreleme algoritmalarının yanında ek olarak, diğer şifreleme algoritmalarına oranla daha hızlı olan DMN şifreleme adı altında yeni bir simetrik dizi (akış) şifreleme algoritması geliştirilmiş ve farklı bir seçenek olarak sunulmuştur.

Sıkıştırılmış ve şifrelenmiş olan veriler ham ses dosyası (wav) üzerine LSB tekniği kullanılarak gizlenmiştir. Bu yöntemle elde edilen stego yapıların algılanamazlık ölçümleri $\chi 2$ (ki-kare) testleri ile yapılmış ve sonuçlarına sonraki bölümde algılanamzlık analizi alt başlığında yer verilmiştir. Gizleme işleminin yapıldığı ses (wav) dosyalarının yapısı bölüm içerisinde alt başlık olarak anlatılmıştır. Ses dosya yapısını daha iyi anlayabilmek için ses dalgası/sinyali ve ses sinyalinin sayısal formata nasıl dönüştürüldüğü konularına da yer verilmiştir. Geliştirilen uygulama ve bu uygulamada kullanılan materyaller alt başlıklarda örneklendirilerek anlatılmıştır.

4.1. Ses Dalgası

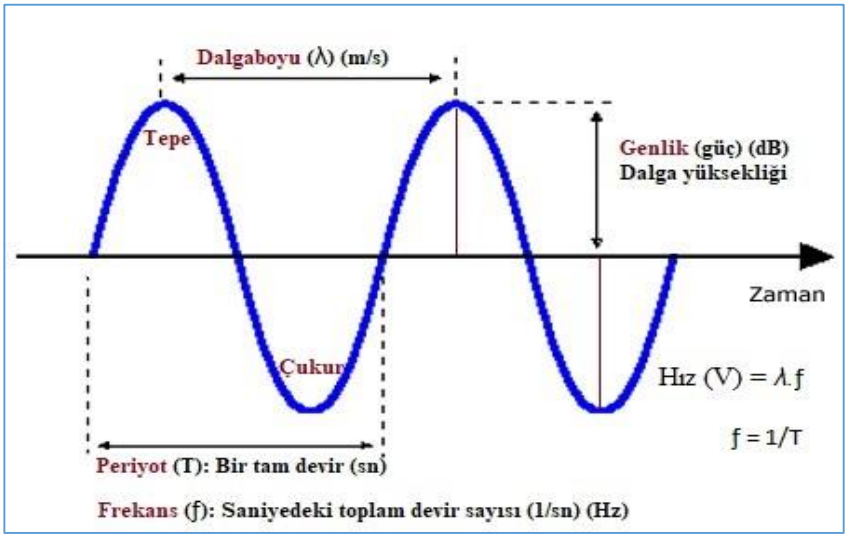
Veri şifreleme ve ses dosyaları üzerinde veri gizleme işlemini gerçekleştiren bu çalışmada kullanılan en önemli materyal olan ses dosylarına ve bir ses dalgasının/sinyalinin nasıl sayısal formata dönüştürüldüğüne bakmakta fayda olacaktır. Bu bakımdan ses nedir, ses dalgasının özellikleri/parametreleri nedir sorularını cevaplamak yerinde olacaktır.

4.1.1. Ses dalgasının yapısı ve özellikleri

Ses; bir maddenin içindeki moleküllerin herhangi bir nedenden dolayı hareket etmesi sonucu ortaya çıkan titreşimlerdir. Şekil 4.1.'de gösterildiği gibi bir ses sinyalinin tanımlanması veya yeniden yapılandırılması için kullanılan üç ana karakteristik özelliği vardır. Bu özellikler; sesin genliği, frekansı ve dalga boyudur [58]. Ses sinyallerin bu özellikleri bu dosyalar üzerinde yapılacak olan steganofrafi uygulamalarında büyük önem taşımaktadır.

Genlik: Ses dalgalarının neden olduğu atmosferik basınçtaki değişim derecesinin bir ölçüsüdür. Bu ölçü ses dalgasının en yüksek (tepe) veya en alçak (çukur) noktasının

denge noktasına olan uzakığıdır. Ses dalgasındaki genlik değeri doğrudan ses yüksekliği ile ilgilidir. Ölçü birimi desibel (dB) olarak ifade edilir. İnsan kulağının ses eşiği 0-120 dB arasındadır.



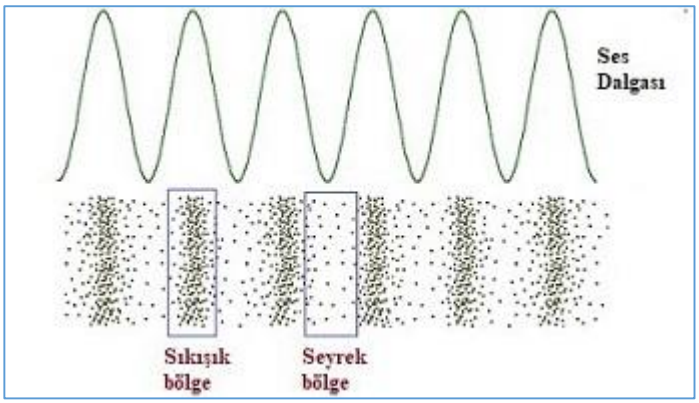
Şekil 4.1. Ses dalgası ve parametreleri

Frekans (f): Ses frekansı, birim zamanda (saniyede) oluşan döngü/devir sayısıdır. Frekans değerinin hesaplanması, ileri geri meydana gelen titreşimlerin zamana bağlı olarak ölçülmesi sonucunda elde edilir. Freakans birimi Hertz (Hz veya s-1) ile temsil edilir ve sesin perdesi ile doğru orantılıdır. Sağlıklı bir insan 20-20000 Hz (titreşim/saniye) arasındaki sesleri işitebilir [58]. Sesin frekansı değeri sadece sesi üreten kaynağa bağlı olarak değişir yani sesin üretildiği ortamdan bağımsızdır. İnce sesleri kalın seslerden ayıran özellik sesin sahip olduğu frekans değeridir. Sesin frekansı arttıkça ses incelirken, frekans değeri düşük olan sesler daha kalındır. Örnek olarak frekansı daha düşük kaplan sesi kedi sesine göre daha kalındır.

Dalga boyu (λ): Bir kaynaktan yayılan periyodik dalgaların ard arda gelen iki tepe ya da iki çukuru arasındaki yatay uzunluğa dalga boyu denir. Dalga boyu λ (lamda) ile gösterilir. Birimi metre/saniye (m/s) ile temsil edilir. Sesin (v) hızına ve frekansına bağlıdır. Dalga boyu, frekans ve ses hızı arasndaki ilişkiyi gösteren formul Denklem 4.1.'de gösterilmiştir [58]. Ses enerjisini ileten taneciklerin sık olduğu (taneciklerin diğer taneciklere enerjilerini aktardığı) bölgeye ses dalgasının tepe noktası veya

dalga tepesi denir. Ses enerjisini ileten taneciklerin seyrek olduğu (enerjiyi aktaran taneciklerin bulunduğu) bölgeye ses dalgasının çukur noktası veya dalga çukuru denir. Ses sinyallerindeki tepe ve çukur noktalarının örneği Şekil 4.2.'de gösterilmiştir.

$$\Lambda = \frac{V}{f} \tag{4.1}$$



Şekil 4.2. Ses sinyalindeki tepe ve çukur noktaların yoğunluğu

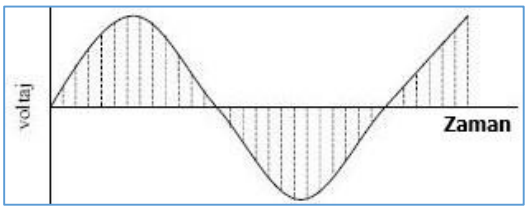
Bu üç temel karakteristik özelliğin dışında Şekil 4.1.'de de yer verildiği gibi ses sinyallerinde/dalgalarında kullanılan bir başka terim ise periyot özelliğidir. *Periyot* Bir ses dalgasının oluşabilmesi için gerekli zamana denir ve *T* ile gösterilir. Birimi zaman türünden (saniye, dakika, saat v.b.) ifade edilmektedir. Ses sinyallerinde periyot, frekans ile ters orantılıdır. Bu orantıyı ve aralarındaki ilişkiyi gösteren formül Denklem 4.2.'de verilmiştir.

$$T.f = 1 \tag{4.2}$$

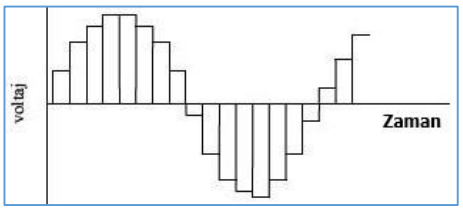
4.1.2. Ses sinyalinin sayısal formata dönüştürülmesi

Ses sinyali sesin "elektriksel formu" şeklinde tanımlanabilir. Ses sinyali, sesin saklanıp yeniden üretebilmesi, kaydedebilmesi ve uzak mesafelere iletebilmesi için

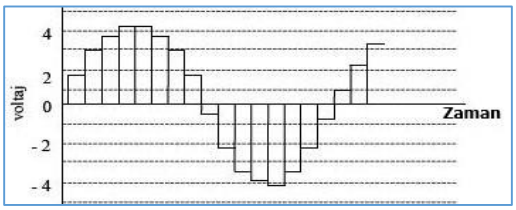
bir "çevirimden geçmiş" halidir. Ses sinyallaeri Dijital ve Analog sinyal olmak üzere iki çeşittir. Analog ses sinyali, kaynaktan çıkan sesin birebir orjinal (tespit edilmiş) biçimidir. Analog ses sinyalleri üzerinden belirli zaman dilimlerinde, belirli örnekler alınıp bu örneklerin sadece 1 ve 0 değerleri ile ifade edilmesiyle elde edilen yeni sinyale ise Dijital ses sinyali denilmektedir. Analog bir ses sinyalinin Dijital ses sinyaline dönüştürme adımları Şekil 4.3.'de gösterildiği gibidir.



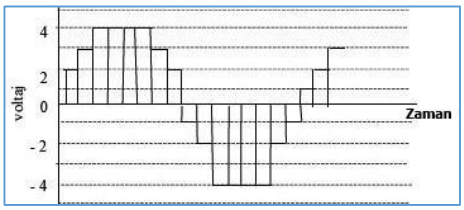
a) Analog sinyal üzerinde sayısal örnekler oluşturulması



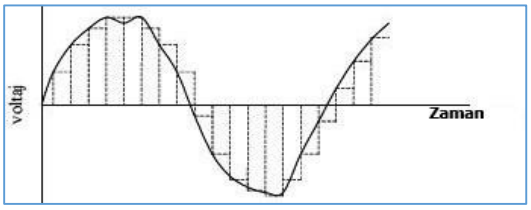
b) Sinyalin örnekleme sonrasındaki durumu



c) Sinyallere sayısal değerler atanması



d) Sayısal değerlerin tam sayıya dönüştürülmesi



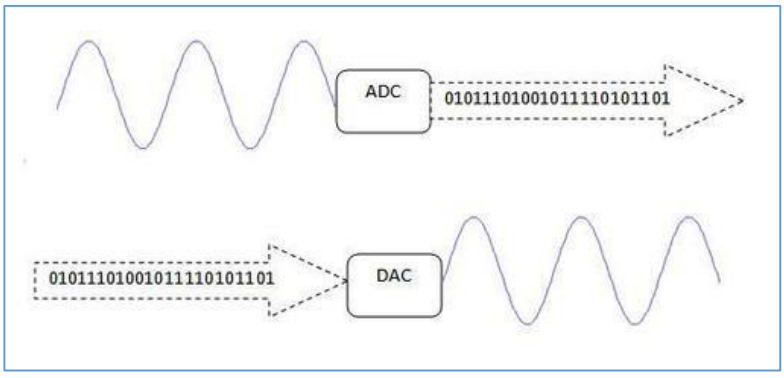
e) Elde edilen sayısal sesin filtreden geçirilmiş hali

Şekil 4.3. Analog sinyalin sayısal formata dönüştürülme adımları [58]

Analog kelime anlamı olarak aslına özdeş demektir. Ses dalgalarında olduğu gibi, ses sinyalinin kayıt ve dinleme sırasında kesintisiz bir bütün olduğunu belirtir. Dijital kelimesi ise sayısal anlamına gelir. Sesin elektronik ortamda sayısal veri şeklinde saklanması ve kullanılmasını ifade eder. Şekil 4.3.'de gösterildiği üzere Dijital ses sinyalleri analog ses sinyallerinden veya doğrudan canlı kayıt sırasında analog-dijital dönüştürücüler (ADC – Analog Digital Convertor) ile çok sayıda komşu dilimlere

bölünerek sayısallaştırılır. Bu dilimlerin çokluğu seste kalitenin yükselmesini sağlar. Dijital ses sinyalleri dinlenmek istediğinde yeniden dijital-analog dönüştürücüler (DAC – Digital Analog Convertor) ile analog elektriksel sinyale dönüştürülür. ADC ve DAC dönüşüm ileyişi Şekil 4.4.'de verilmiştir.

Sesin sayısal ortama aktarılma işlemi, bir resmin sayısallaştırlması işlemine benzemektedir. Bir resim dosyası üzerinde bulunan piksel sayısı ses dosyalarındaki örnekleme hızına karşılık gelmektedir. Ses dosyalarındaki örnekleme hızı ise Analog ses sinyalleri üzerinde birim zamanda elde edilen ses örneği sayısı olarak karşımıza çıkmaktadur. Ses dosyalarının sahip olduğu ayrıntı, içersinde barındırdığı örnekleme hız sayısına bağlı olarak değişir. Bu değişim göstergesi bir resim dosyanın sahip olduğu çözünürlük (piksel) sonucu ortaya koyduğu ayrıntıya eş değerdir. Seslerdeki durum da bunun gibidir, örnekleme hızı arttığında sesin barındırabileceği ayrıntılar artacaktır. Öte yandan, ne kadar piksel yoğunluğuna sahip olursa olsun, düşük renk derinliğindeki resim, canlı ve gerçekçi olamayacaktır. Aynı durum, sayısal sesler için de geçerlidir ve ses ne kadar yüksek örnekleme hızına sahip olursa olsun düşük bit derinliğine sahipse, sahip olduğu tonlar yetersiz kalacak, canlı ve gerçekçi olamayacaktır.

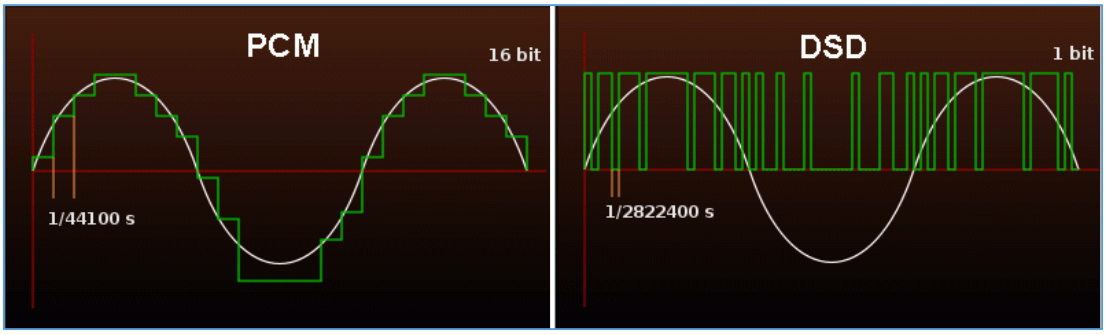


Şekil 4.4. ADC ve DAC dönüştürme

Ses sinyallerini sayısallaştırmada kullanılan iki temel teknik vardır. Bunlar PCM (Pulse Code Modulation - Darbe Kod Modülasyonu) ve DSD/DSM (Direct Stream Digital/Delta Sigma Modulation) teknikleridir.

PCM (Darbe Kod Modülasyonu - Pulse Code Modulation): Analog sinyalin sayısal formata dönüştürülmesinde kullanılan en yaygın yoldur. PCM, sıkıştırmasız bir yapı sunar ve ses dalgasını orijinal haliyle sayısal formata getirme mantığı üzerine tasarlanmıştır. Disk üzerinde çok yer kaplamasının sebebi sıkıştırma işleminin uygulanmamasıdır. Ses işlemcisi (ses kartı) üzerimde basitçe çalıştırmak mümkündür. PCM diğer dijital ses formatlarının temelini oluşturur ve örnekleme hızı ve bit derinliği kavramlarına dayanır. Analog sinyal üzerinden birim zamanda belirli örnekler alınır, her örnek 0 ve 1 ile ifade edilerek bir kodlamaya tabi tutulur. Örneğin CD'ye yazılacak bir analog sesten, saniyede 44100 kere örnek alınır (örnekleme frekansı). Her örnek 16 bit'lik bir sayıya çevrilir. Böyle bir kayıttan kısaca 44.1kHz/16bit olarak ifade edilir. PCM ses kalitesini arttırmak için örnekleme frekansının ve bit uzunluğunun arttırılması gereklidir. PCM tekniği ile yapılan dönüşümlerde 44.1kHz/16bit - 384kHz/32bit değerlerine sahip ses sinyalleri elde edilebilir [71].

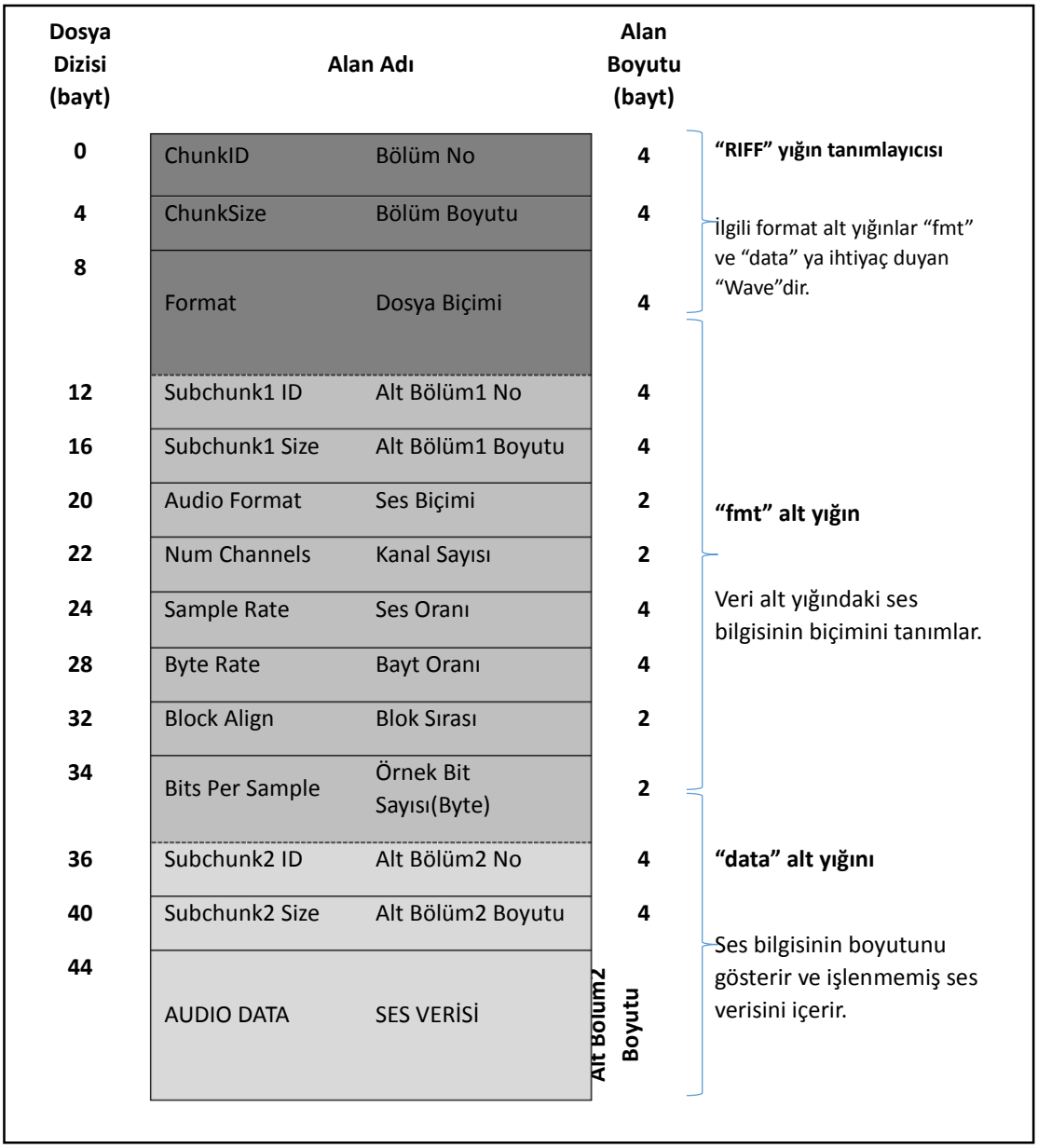
DSD (Direct Stream Digital): DSD Sony ve Philips tarafından üretilmiş olan süper ses CD'si (SACD) için ses sinyallerini sayısallaştıran bir tekniktir. DSD, darbe yoğunluklu modülasyon kodlamasını kullanır. PCM tekniğine göre çok daha fazla örnekleme frekansına sahiptir. PCM'de analog bir sinyalin CD ortamı için 44.100 kere örneklenip 16 bit uzunluğunda bir sayıya çevrilmektedir. Bu saniyede 705.600 bitlik bir veri hızı demektir. DSD'de ise analog sinyalden saniyede 2.822.400 örnek alınır ve bu 1 bit uzunluğunda (0 ya da 1) bir sayıya çevrilir. Elde edilen bu sayısal ses 2822.4Mhz/1bit şeklinde ifade edilir. Şekil 4.5.'de bir sinüs sinyalinin, saniyede 44.100 kere örneklendiği ve 16 bitlik bir sayıya çevrildiği PCM ve saniyede 2.842.400 kere örneklendiği ve 1 bit DSD olarak ifade edilişi gösterilmiştir. [71].



Şekil 4.5. Ses sinyalinin PCM ve DSD tekniklerine göre sayısal formata dönüştürülmesi [71]

4.2. WAV Dosya Yapısı

WAV dosyaları analog ses verilerine sıkıştırma işlemi uygulanmadan PCM (Pulse Code Modulation) yöntemiyle elde edilen sayısal ses dosyalarıdır. Bu bakımdan analog seslere daha yakın bir ses kalitesi sunmaktadır. Sıkıştırma yapılmadığı için disk üzerinde fazla yer kaplarlar. Bu tip bir dosya yapısının ilkel versiyonu ise Microsoft‟un “.riff” (Resource Interface File Format) uzantılı dosya yapısıdır. İşlenmemiş yani sıkıştırılmamış bir yapıya sahip olan wav dosyaları veri gizleme işlemlerinde taşıyıcı nesne olarak kullanılabilecek ideal ses dosyalarındandır.



Şekil 4.6. WAV dosya yapısı [72]

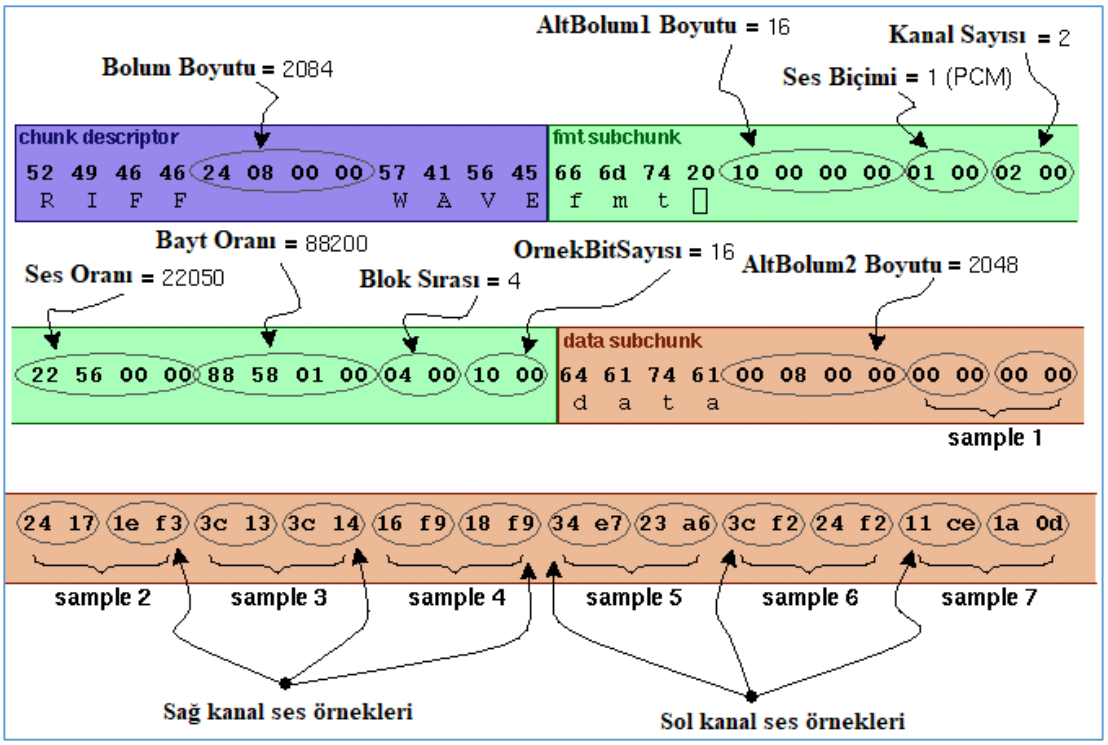
Tablo 4.1. WAV dosya yapısı yığın açıklamaları [72]

Yığın	Boyut (Bayt)	Başlık	Açıklama
RIFF	4	ChunkID (Bölüm No)	ASCII biçimindeki "RIFF" yazısını içerir. ChunkID kısmında "RIFF" kelimesi yazılıdır. RIFF WAV dosyalarını da içeren genel dosya formatının ismidir.
	4	ChunkSize (Bölüm Boyutu)	Yığın boyutunu içerir. ChunkSize alanındaki değer, kendisinden sonra gelen tüm bölümlerin toplam boyutunu bayt cinsinden belirtmektedir.
	4	Format (Dosya Biçimi)	ASCII biçiminde "WAVE" bilgisini içerir. Bu bilgiler bu dosyanın bir WAV dosyası olduğuna işaret etmektedirler.
fmt	4	Subchunk1ID (Alt Bölüm1 No)	"fmt" yazısını içerir.
	4	Subchunk1Size (Alt Bölüm1 Boyutu)	PCM için 16'dir. Kendisiyle ubchunk2ID alanı arasında kalan bölümlerin toplam boyutunun ne kadar bayt olduğunu göstermektedir.
	2	Audioformat (Ses Biçimi)	PCM için 1‟dir. Sıkıştırma tiplerinin bazıları için 1‟den faklı değerler mevcuttur.
	2	NumChannels (Kanal Sayısı)	NumChannels değeri kaç kanal olduğunu gösterir. Mono için 1, Stereo için 2 değerini alır.
	4	SampleRate (Ses Oranı)	8000, 44100 gibi örnek oranlarıdır.
	4	ByteRate (Bayt Oranı)	= SampleRate x NumChannels x BitsPerSample/8
	2	BlockAlign (Blok Sırası)	= NumChannels x BitsPerSample/8
	2	BitsPerSample (Örnek Bit Sayısı)	BitsPerSample verisi bir ses örneğinin kaç bitlik bir değere sahip olduğunu göstermektedir. WAV dosyaları için bu değer 8 bit (1 byte) veya 16 bit (2 byte) olmaktadır.
data	4	SubChunk2ID (Alt Bölüm2 No)	"data" yazısını içerir.
	4	SubChunk2Size (Alt Bölüm2 Boyutu)	= NumSamples x NumChannels x BitsPerSample/8 Bu bilgi, data bölümünde bulunan ses verilerinin toplam boyutudur.
	N	Data (ses örnekleri)	Ses verileri (44.bayttan itibaren).

WAV ses dosyasının yapısı Şekil 4.6.'da gösterildiği gibi, "RIFF" yığın tanımlayıcısı adı verilen başlık bölümü yeralmaktadır. Sonraki bölümde ise "WAVE" veri yığını gelmektedir. Bu yığın kendi içerisinde iki alt bölüm barındırmaktadır.

WAVE veri yığının alt veri parçalarından birincisi "fmt" veri parçası olup mevcut dosyanın veri biçimiyle ilgili bilgileri içermektedir. Diğer alt veri parçası olan "data" ise asıl ses örnekleri verisinden oluşmaktadır [72]. Şekil 4.6'da verilen WAV dosya

yapısı içerisinde bulunan bölüm ve alt bölümlerin açıklamaları Tablo 4.1.'de verilmiştir.



Şekil 4.7. Örnek WAV dosyası[72]

Önceki paragraflarda bahsedildiği üzere ses dosyalarında "data" alt yığını asıl ses örneklerini (audio samples) içermektedir. Steganografi işlemleri bakımından veri gizleme yaklaşımlarının uygulandığı alanı temsil etmektedir. Gizlenmek istenilen mesaj bu alan ("data") içerisindeki LSB bitleri üzerinde işlenmektedir. Sayısal ses örnekleri 8 veya 16 bit olmak üzere iki farklı uzunluğa sahip olabilirler. Bu seslerden 16 bitlik ses örneklerinden oluşan ses dosyalarına stereo (çift kanal), 8 bitlik ses örneklerinden oluşan WAV dosyalarına mono (tek kanal) ses dosyaları denir. Stereo ses dosyalarında bilgi iki kanalla çoklanır ve daha kaliteli ses çıkışı sağlanır. Mono ses dosyalarında en fazla 128 adet kategori oluşurken, stereo ses dosyalarında 32768 adet kadar kategori oluşabilmektedir.

Örnek olarak, onaltılık sayı tabanında (hexadecimal) verilmiş olan 72 bayt büyüklüğünde bir WAVE dosyasının verilerinin kodlanmış hali şöyle olsun:

```
52 49 46 46 24 08 00 00 57 41 56 45 66 6d 74 20 10 00 00 00 01 00 02 00
22 56 00 00 88 58 01 00 04 00 10 00 64 61 74 61 00 08 00 00 00 00 00 00
24 17 1e f3 3c 13 3c 14 16 f9 18 f9 34 e7 23 a6 3c f2 24 f2 11 ce 1a 0d
```

Verilen bu ses örneğinin Şekil 4.6. ve Tablo 4.1.'e göre oluşacak olan bölüm ve alt bölümlerinin yapısı Şekil 4.7.'da gösterildiği gibi olacaktır.

4.3. ASCII Karakter Tablosu

Günümüz bilgisayar sistemlerinde, bir bilginin bilgisayar hafızasında temsil edilebilmesi için ikili (binary) sayısal karşılığının olması gerekmektedir. ASCII ve benzeri kodlama sistemleri de sayısal ve sayısal olmayan karakterlere ve sembollere, belirli sayısal değerler atayarak bilgisayar sistemlerinde temsil edilebilmesini sağlamaktatır. Özetle bu ve benzeri kodlama sistemleri bilgisayarın işlediği ikili sayılarla (0,1) oluşturulmuş olan sinyalleri, insanların anlayabileceği sembollere çevirmek için kullanılır. ASCII tablosu bir Amerikan standardı olup Latin alfabelerinden olan İngilizce temel alınarak geliştirilmiş ve içinde Türkçe karakterleri barındırmayan bir kodlama sistemidir. ASCII ilk olarak 1963 yılında ANSI tarafından (American National Standards Institute - Amerikan Ulusal Standartlar Enstitüsü) tarafından ilk kez kullanılmıştır [65].

Standart ASCII tablosu içerisinde toplamda 128 karakter vardır. İkili sayı sisteminde bu karakterler 7 adet bit (27) ile ifade edilebilir. Standart ASCII karakter tablosu ikili, sekizli, onlu ve onaltılı değerleri ile beraber Şekil 4.8.'da gösterildiği gibidir. Bu tabloda 33 karakter yazdırılamayan kontrol kodları ve yazıcılar gibi çevre birimlerini kontrol etmek için kullanılır, diğer 95 karakter ise basılabilen sembolleri ve karakterleri göstermektedir [65]. Yazdırılmayan karakter kodlarının açıklamaları Tablo 4.2.'de verilmiştir.

Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char
0	0x00	000	0000000	NUL	32	0x20	040	0100000	space	64	0x40	100	1000000	@	96	0x60	140	1100000	`
1	0x01	001	0000001	SOH	33	0x21	041	0100001	!	65	0x41	101	1000001	A	97	0x61	141	1100001	a
2	0x02	002	0000010	STX	34	0x22	042	0100010	"	66	0x42	102	1000010	B	98	0x62	142	1100010	b
3	0x03	003	0000011	ETX	35	0x23	043	0100011	#	67	0x43	103	1000011	C	99	0x63	143	1100011	c
4	0x04	004	0000100	EOT	36	0x24	044	0100100	$	68	0x44	104	1000100	D	100	0x64	144	1100100	d
5	0x05	005	0000101	ENQ	37	0x25	045	0100101	%	69	0x45	105	1000101	E	101	0x65	145	1100101	e
6	0x06	006	0000110	ACK	38	0x26	046	0100110	&	70	0x46	106	1000110	F	102	0x66	146	1100110	f
7	0x07	007	0000111	BEL	39	0x27	047	0100111	'	71	0x47	107	1000111	G	103	0x67	147	1100111	g
8	0x08	010	0001000	BS	40	0x28	050	0101000	(	72	0x48	110	1001000	H	104	0x68	150	1101000	h
9	0x09	011	0001001	TAB	41	0x29	051	0101001	)	73	0x49	111	1001001	I	105	0x69	151	1101001	i
10	0x0A	012	0001010	LF	42	0x2A	052	0101010	*	74	0x4A	112	1001010	J	106	0x6A	152	1101010	j
11	0x0B	013	0001011	VT	43	0x2B	053	0101011	+	75	0x4B	113	1001011	K	107	0x6B	153	1101011	k
12	0x0C	014	0001100	FF	44	0x2C	054	0101100	,	76	0x4C	114	1001100	L	108	0x6C	154	1101100	l
13	0x0D	015	0001101	CR	45	0x2D	055	0101101	-	77	0x4D	115	1001101	M	109	0x6D	155	1101101	m
14	0x0E	016	0001110	SO	46	0x2E	056	0101110	.	78	0x4E	116	1001110	N	110	0x6E	156	1101110	n
15	0x0F	017	0001111	SI	47	0x2F	057	0101111	/	79	0x4F	117	1001111	O	111	0x6F	157	1101111	o
16	0x10	020	0010000	DLE	48	0x30	060	0110000	0	80	0x50	120	1010000	P	112	0x70	160	1110000	p
17	0x11	021	0010001	DC1	49	0x31	061	0110001	1	81	0x51	121	1010001	Q	113	0x71	161	1110001	q
18	0x12	022	0010010	DC2	50	0x32	062	0110010	2	82	0x52	122	1010010	R	114	0x72	162	1110010	r
19	0x13	023	0010011	DC3	51	0x33	063	0110011	3	83	0x53	123	1010011	S	115	0x73	163	1110011	s
20	0x14	024	0010100	DC4	52	0x34	064	0110100	4	84	0x54	124	1010100	T	116	0x74	164	1110100	t
21	0x15	025	0010101	NAK	53	0x35	065	0110101	5	85	0x55	125	1010101	U	117	0x75	165	1110101	u
22	0x16	026	0010110	SYN	54	0x36	066	0110110	6	86	0x56	126	1010110	V	118	0x76	166	1110110	v
23	0x17	027	0010111	ETB	55	0x37	067	0110111	7	87	0x57	127	1010111	W	119	0x77	167	1110111	w
24	0x18	030	0011000	CAN	56	0x38	070	0111000	8	88	0x58	130	1011000	X	120	0x78	170	1111000	x
25	0x19	031	0011001	EM	57	0x39	071	0111001	9	89	0x59	131	1011001	Y	121	0x79	171	1111001	y
26	0x1A	032	0011010	SUB	58	0x3A	072	0111010	:	90	0x5A	132	1011010	Z	122	0x7A	172	1111010	z
27	0x1B	033	0011011	ESC	59	0x3B	073	0111011	;	91	0x5B	133	1011011	[	123	0x7B	173	1111011	{
28	0x1C	034	0011100	FS	60	0x3C	074	0111100	<	92	0x5C	134	1011100	\	124	0x7C	174	1111100	|
29	0x1D	035	0011101	GS	61	0x3D	075	0111101	=	93	0x5D	135	1011101	]	125	0x7D	175	1111101	}
30	0x1E	036	0011110	RS	62	0x3E	076	0111110	>	94	0x5E	136	1011110	^	126	0x7E	176	1111110	~
31	0x1F	037	0011111	US	63	0x3F	077	0111111	?	95	0x5F	137	1011111	_	127	0x7F	177	1111111	DEL

Şekil 4.8. Standart ASCII tablosu [65]

Bu sebeple Tükçe gibi birçok dile ait karakterleri içermediği için, toplam da 256 karakter içeren ve sekiz bit (28) ile ifade edilebilen Genişletilmiş ASCII tabloları oluşturulmuştur. Genişletilmiş ASCII tablosu Şekil 4.9.'da verilmiştir. Tabloda verilen ilk 128 karakter Standart ASCII sistemi ile aynıyken 128 inci karakterden sonrası ülkelerin kullandığı dile göre düzenlenmiştir. Genişletilmiş ASCII tablosundan bahsedilirken hangi karakter kümesine göre genişletildiğinin belirtilmesi gerekir. Örneğin Batı Avrupa dillerindeki karakterleri temsil etmek için kullanılan ISO-8859-1 karakter kümesi bugün yaygın bir şekilde kullanılmaktadır. Türkçe'ye has olan kodlama sistemi ise ISO-8859-9 serisidir. Yalnızca Türkçe'de (ve aynı gruptan kimi Türk dillerinde) bulunan "ğ,ş,ı, vb." harfleri yalnızca bu kodlama sisteminde bulunur.

0	<NUL>	32	<SPC>	64	@	96	`	128	Ä	160	†	192	¿	224	‡
1	<SOH>	33	!	65	A	97	a	129	Å	161	°	193	¡	225	·
2	<STX>	34	"	66	B	98	b	130	Ç	162	¢	194	¬	226	‚
3	<ETX>	35	#	67	C	99	c	131	É	163	£	195	√	227	„
4	<EOT>	36	$	68	D	100	d	132	Ñ	164	§	196	ƒ	228	‰
5	<ENQ>	37	%	69	E	101	e	133	Ö	165	•	197	≈	229	Â
6	<ACK>	38	&	70	F	102	f	134	Ü	166	¶	198	Δ	230	Ê
7	<BEL>	39	'	71	G	103	g	135	á	167	ß	199	«	231	Á
8	<BS>	40	(	72	H	104	h	136	à	168	®	200	»	232	Ë
9	<TAB>	41	)	73	I	105	i	137	â	169	©	201	…	233	È
10	<LF>	42	*	74	J	106	j	138	ä	170	™	202		234	Í
11	<VT>	43	+	75	K	107	k	139	ã	171	´	203	À	235	Î
12	<FF>	44	,	76	L	108	l	140	å	172	¨	204	Ã	236	Ï
13	<CR>	45	-	77	M	109	m	141	ç	173	≠	205	Õ	237	Ì
14	<SO>	46	.	78	N	110	n	142	é	174	Æ	206	Œ	238	Ó
15	<SI>	47	/	79	O	111	o	143	è	175	Ø	207	œ	239	Ô
16	<DLE>	48	0	80	P	112	p	144	ê	176	∞	208	–	240	
17	<DC1>	49	1	81	Q	113	q	145	ë	177	±	209	—	241	Ò
18	<DC2>	50	2	82	R	114	r	146	í	178	≤	210	"	242	Ú
19	<DC3>	51	3	83	S	115	s	147	ì	179	≥	211	″	243	Û
20	<DC4>	52	4	84	T	116	t	148	î	180	¥	212	'	244	Ù
21	<NAK>	53	5	85	U	117	u	149	ï	181	µ	213	′	245	ı
22	<SYN	54	6	86	V	118	v	150	ñ	182	∂	214	÷	246	ˆ
23	<ETB>	55	7	87	W	119	w	151	ó	183	∑	215	◊	247	˜
24	<CAN>	56	8	88	X	120	x	152	ò	184	∏	216	ÿ	248	¯
25	<EM>	57	9	89	Y	121	y	153	ô	185	π	217	Ÿ	249	˘
26	<SUB>	58	:	90	Z	122	z	154	ö	186	∫	218	⁄	250	˙
27	<ESC>	59	;	91	[	123	{	155	õ	187	ª	219	€	251	°
28	<FS>	60	<	92	\	124	|	156	ú	188	º	220	‹	252	¸
29	<GS>	61	=	93	]	125	}	157	ù	189	Ω	221	›	253	˝
30	<RS>	62	>	94	^	126	~	158	û	190	æ	222	fi	254	˛
31	<US>	63	?	95	_	127	<DEL>	159	ü	191	ø	223	fl	255	ˇ

Şekil 4.9. Genişletilmiş ASCII tablosu [65]

Tablo 4.2. Yazdırılmayan ASCII kodlarının anlamları

KOD	EN	TR	KOD	EN	TR
ACK	Acknowledge	Alındı	**HT**	Horızontal tabulatıon	Yatay tablo
BEL	Bell or alarm	Alarm	**IS**	Informatıon separatıon	Bilgi ayırıcı
BS	Backspace	Geriye alma	**LF**	Lıne feed	Satır iletme
CAN	Cancel	Kesme	**NAK**	Negatıve acknowledge	Alınmadı
CR	Carrıage return	Satırbaşı	**NL**	New lıne	Yeni satır
DC	Device control	Düzen kontrol	**NUL**	Nul or all zeroes	Nul veya sıfır
DEL	Delete	Silme	**RS**	Record separator	Kayıt ayırıcı
DLE	Data lınk escape	Veri hattı merdiveni	**SI**	Shıft ın	İçeri kaydırma
EM	End of medıum	Ortam sonu	**SO**	Shıft out	Dışarı kaydırma
ENQ	Enquıry	Sorgu soruşturma	**SOH**	Start of headıng	Mesaj başı
EOT	End of transmıssıon	İletim sonu	**SPC**	Space	Boşluk
ESC	Escape	Kaçma	**STX**	Start of text	Metin başı
ETB	End of transmıssıon block	İletim blou sonu	**SUB**	Substıtute	Yerine koyma
ETX	End of text	Metin sonu	**SYN**	Synchronous ıdle	Eş zamanlı boşluk
FE	Format effector	Biçim etkileyici	**TC**	Transmıssıon control	İletim kontrolu
FF	Form feed	Sayfa ilerletme	**US**	Unıt separator	Birim ayırıcı
FS	Fıle separator	Kütük ayırıcı	**VT**	Vertıcal tabulatıon	Düşey tablo
GS	Group separator	Grup ayırıcı			

4.4. Veri Sıkıştırma Teknikleri ve LZW Algoritması

Veri sıkıştırma işlemi, belirli uzunluktaki verilerin çeşitli yöntemlerle daha az bellek kullanılması amacıyla geliştirilmiştir. Bu sayede bellek üzerinde yer tasarrufu, veri aktarımında da zaman tasarrufu yapılabilmektedir.

Özellikle steganografi işlemlerinde kullanılmak üzere birçok veri sıkıştırma yöntemi geliştirilmiş ve bunlar daha sonra uygulamalarda ve akademik çalışmalarda test edilerek sonuçlar gözlemlenmiştir [66,67,68]. Bu yöntemler genel olarak 4 temel sınıfta toplanabilir [44].

1. Tekrarlama Sayısı Kodlama (Run Length Coding).
2. İstatiksel Yöntemler (Statistical Methods).
3. Yer Değiştirmeye Dayalı Yöntemler (Transforms).
4. Sözlük-Tabanlı Yöntemler (Dictionary-Based Methods).

Yer değiştirme sıkıştırması, tekrar eden karakterlerin tümü için daha kısa bir ifade kullanılmasını içermektedir. İstatistiksel sıkıştırma teknikleri, karakterlerin hesaplanan olasılıklarına göre en kısa ortalama kod uzunluğunun üretimini içermektedir. Bu tür sıkıştırma tekniklerinde, kaynak dosyadaki karakterler ikili koda dönüştürülür. Dosyadaki en genel karakterler en kısa ikili kodu alırken, en az genel olan karakterler en uzun ikili kodu almaktadır. Sözlük tabanlı sıkıştırma metotlarında ise, metindeki karakterler, oluşturulan bir sözlüğe göre indis veya işaretçi kodu ile gösterilirler (Lempel Ziv Welch - LZW). Çoğu sıkıştırma algoritması, sıkıştırma oranını artırmak için farklı veri sıkıştırma tekniklerinin kombinasyonlarını kullanmaktadır [30]. Bu ve benzeri teknikler kullanarak gerçekleştirilen sıkıştırma işlemleri sonucunda elde edilen sıkıştırma miktarı, sıkıştırma oranı olarak bilinen ve Denklem 4.3.'de gösterildiği gibi bir C faktörü ile ölçülmektedir [69, 30].

$$C = \frac{So}{Sc} \tag{4.3}$$

Denklemde kullanılan So; orijinal dosya boyutunu, Sc ise sıkıştırılan dosya boyutunu temsil etmektedir. Sıkıştırma oranı C:1 şeklinde ifade edilmektedir. Sıkıştırma miktarı aynı zamanda Denkelem 4.4.'de gösterilmiş olduğu üzere R ile yüzde (%) şeklinde ifade edilen orijinal veri miktarındaki azalma ile de ölçülebilmektedir [69, 30]:

$$R = \frac{So - Sc}{So} \tag{4.4}$$

Bu uygulamada Sözlük Tabanlı yöntemlerden olan LZW algoritmasından faydalanılmış ve buna bağlı olarak iletilmek istenen gizli mesaj yeniden kodlanmıştır. LZW kodlama tekniğinde hali hazırda sabit bir sözlük yoktur. Ancak karşı/alıcıc tarafa kendi oluşturmuş olduğumuz sözlüğü göndermek zorunda kalmamak için, bu uygulamada yapılmış olduğu gibi temel sözlük olarak ASCII tablosu kullanılabilir. LZW algoritması öncelikle veriyi okur ve sözlükte kodlanan bir diziden yararlanarak mümkün olduğunca büyük veri biti serisi ile eşleşme yapmaya çalışır. Eşleşen veri sırası ve bunu izleyen karakter sonraki veri serilerinin kodlanması amacıyla birlikte gruplandırılarak sözlüğe eklenir. Daha küçük, sıkıştırılmış bir kod daha yüksek sıkıştırma oranıyla sonuçlanırken, sözlük boyutunu da sınırlandırmaktadır[30].

LZW algoritmasının çalışma mantığı ve algoritma adımları aşağıdaki gibidir [30, 70]:

- C veri dizisindeki bir sonraki karakter olsun.
- P + C dizisi sözlükte var mı?
    1. Eğer var ise, P ← P + C (P'yi C ile genişlet)
    2. Yok ise
        - P kod kelimesini, kod dizisine çıkış olarak ver
        - P + C dizisini sözlüğe ekle
        - P ←C (P bu durumda sadece C karakterini içermektedir.)
- Veri dizisinin sonuna gelindi mi?

1. Hayır, ise 2. Adıma git
2. Evet, ise P kod kelimesini, kod dizisine çıkış olarak ver [30].

Verilen bu algoritma adımlarına göre kodlamak istediğimiz "TOBEORNOTTOBE" metninin, ASCII tablosu temel sözlük alınarak, LZW algoritmasına göre nasıl kodlandığı ve kodlama sonucu oluşan yeni sözlük Tablo 4.3.'de gösterilmektedir.

Tablo 4.3. LZW algoritması kodlama örneği

Giriş	Sözlükte Varmı?	Yeni Kod Girişi	Çıkış	Giriş	Sözlükte Varmı?	Yeni Kod Girişi	Çıkış
T	E			NO	H	262-NO	78(N)
TO	H	256-TO	84(T)	O	E		
O	E			OT	H	263-OT	79(O)
OB	H	257-OB	79(O)	TT	H	264-TT	84(T)
B	E			T	E		
BE	H	258-BE	66(B)	TO	E		
E	E			O	E		
EO	H	259-EO	69(E)	OB	E		
O	E			TOB	H	265-TOB	256(TO)
OR	H	260-OR	79(O)	B	E		
R	E			BE	E		
RN	H	261-RN	82(R)	BE, eof	H		258(BE)
N	E						

Bu işlem sonucunda elde edilen mesajın kodlanmış hali; 84 – 79 – 66- 69 – 79 – 82 – 78 – 79 – 84 – 256 – 258 olacak şekilde kodlanmış ve karakter sayısı onüç (13) değerinden onbir (11) değerine indirgenmiştir.

4.5. Geliştirilen Uygulama

Gerçekleştirilen uygulama temel olarak iki kısımdan oluşmaktadır, bu iki temel kısımda kendi içerisinde üç ayrı işlem adımlarını barındırmaktadır. Temel modüllülerden birincisi gönderilmek istenilen mesajı sıkıştırma (compress), şifreleme (encrypt) ve elde edilen kodlanmış verilerin taşıyıcı dosya (R:Bmp, png, gif; S:wav) içerisine gizleme (hide) adımlarını kapsamaktadır. Uygulamanın ikinci kısmı ise gömülü (saklı) veriyi stego dosyadan çıkarma (extract), şifrelenmiş mesajı çözme (decrypt) ve sıkıştırılmış veriyi açma (decompress) işlemlerini gerçekleştirilmektedir. Bu iki temel modül dışında, yapılan işlemlerin (sıkıştırma, şifreleme, gizleme) işlem sürelerini, süreçlerini kaydeden, işleme tabi tutulan taşıyıcı

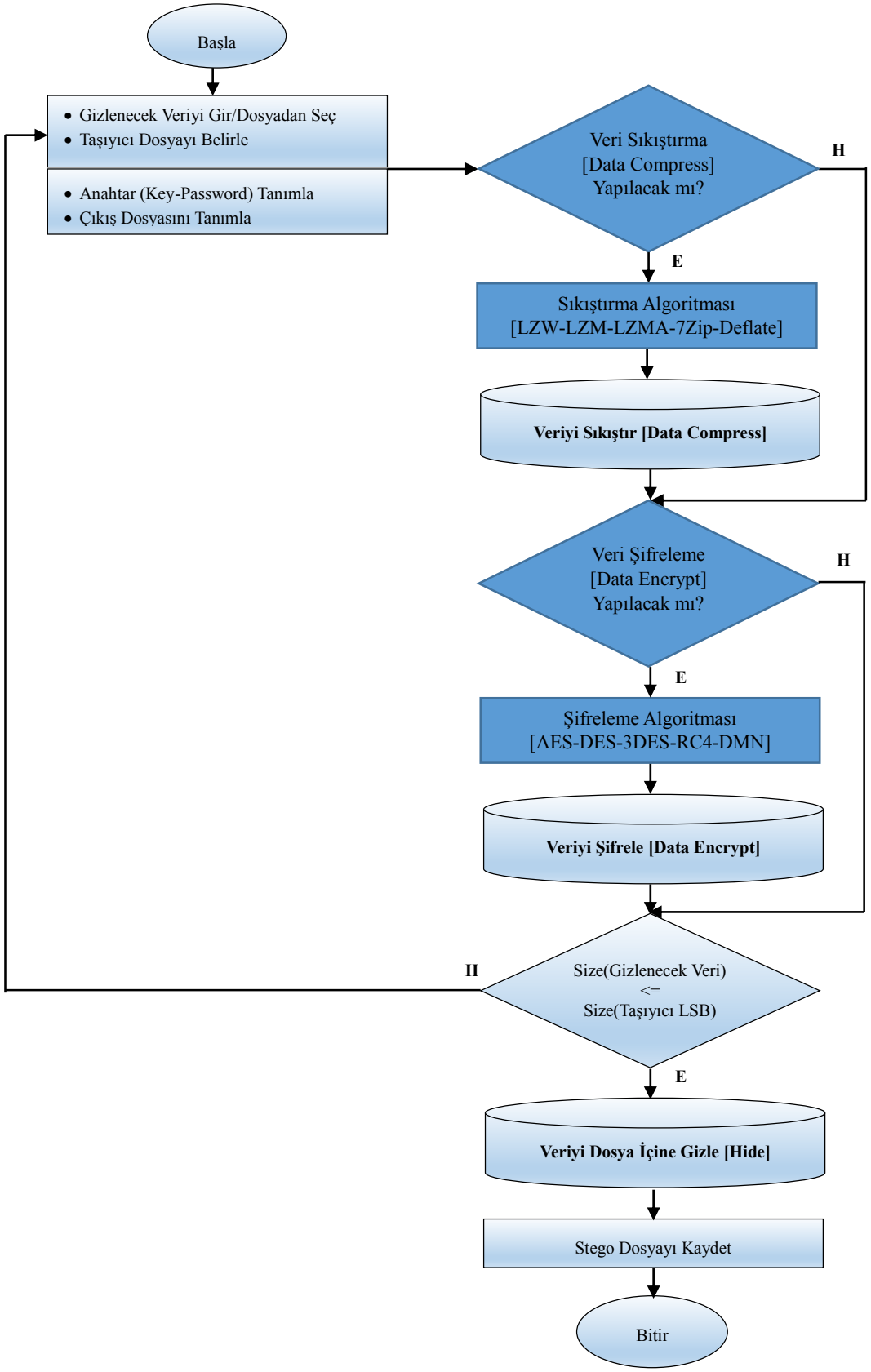
ve stego nesnelerin karşılaştırırılarak analiz edildiği bir üçüncü ara yüz de yer almaktadır.

Gerçekleştirilen uygulama kullanıcı isteğine bağlı olarak yedi farklı şekilde (amaçla) kullanılabilir. Uygulamanın sunduğu işlem seçenekleri aşağıda maddler halinde verilmiştir;

- Sadece veri sıkıştırma [Data compress]
- Sadece veri şifreleme [Data encryption]
- Sadece veri gizleme [Data hide]
- Veri sıkıştırma ve şifreleme [Data compress and encryption]
- Veri sıkıştırma ve gizleme [Data compress and hide]
- Veri şifreleme ve gizleme [Data encryption and hide]
- Veri sıkıştırma, şifreleme ve gizleme [Data compress, encryption and hide]

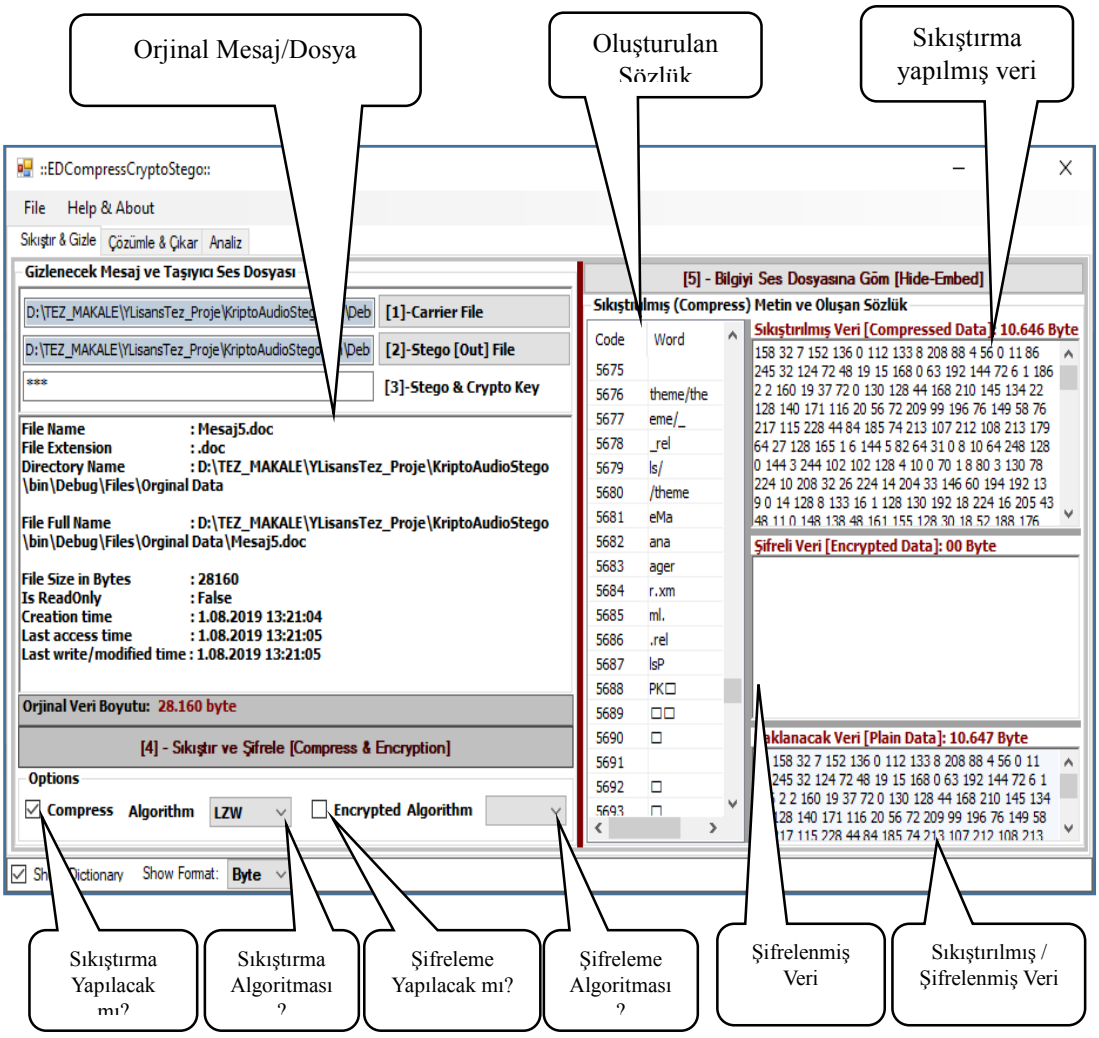
4.5.1. Veri kodlama ve ses dosyası içerisine gizleme

Gerçekleştirilen uygulamada iletilmek istenen gizli mesaj/dosya kullanıcıdan alındıktan sonra kayıpsız veri sıkıştırma algoritmalarından biri olan LZW veri sıkıştırma algoritması temeline dayalı olarak geliştirilen LZW, LZM, LZMA, 7Zip, Deflate algoritmalarından biri kullanılarak yeni bir sözlük elde edilmesi amaçlanmıştır. Bu sayede gönderilmek istenilen bilginin uzunluğu (boyutu) küçültülmüştür. Bu sıkıştırma işlemi ile steganografi temellerinden birisi olan taşıyıcı dosyadaki kapasite sorunun aşılmasına yardımcı olunmuştur. Sıkıştırma işlemi ile elde edilen yeni veri isteğe bağlı olarak kullanıcının seçimi ile belirleyeceği veri şifreleme algoritmalardan (AES-DES-3DES-RC4) biriyle şifrelenebilmektedir. Böylelikle steganografi temellerinden bir diğeri olan sağlamlık adımı başarıyla gerçekleştirilmesi amaçlanmıştır. Yine bu amaçla uygulanan veri gömme algoritması ile saklamak istediğimiz mesaja ait veriler ses dosyasının en az önemli bitlerine sıralı olarak değil rastgele (karmaşık düzen içerisinde) yerleştirilmiştir. Böylelikle yapılacak olan steg-analiz ataklarına karşı algılanamazlık ilkesi yerine getirilmiştir.



Şekil 4.10. Veri sıkıştırma, şifreleme, gizleme işlem adımları

Şekil 4.10.'da geliştirilen uygulamanın veri sıkıştırma, şifreleme ve ses dosyası içerisine gizleme işlemlerinin gerçekleştirildiği iş akış adımları gösterilmiş ve Şekil 4.11.'de ise bu işlemlerin gerçekleştirildiği uygulama arayüzüne yer verilmiştir. Arayüzde görüldüğü gibi gönderilmek istenilen veri (herhangi bir format türünde olabilir) bir dosyasından seçilebileceği gibi, el ile de veri girişi yapılabilir.



Şekil 4.11. Uygulama veri sıkıştırma, şifreleme ve gizleme arayüzü

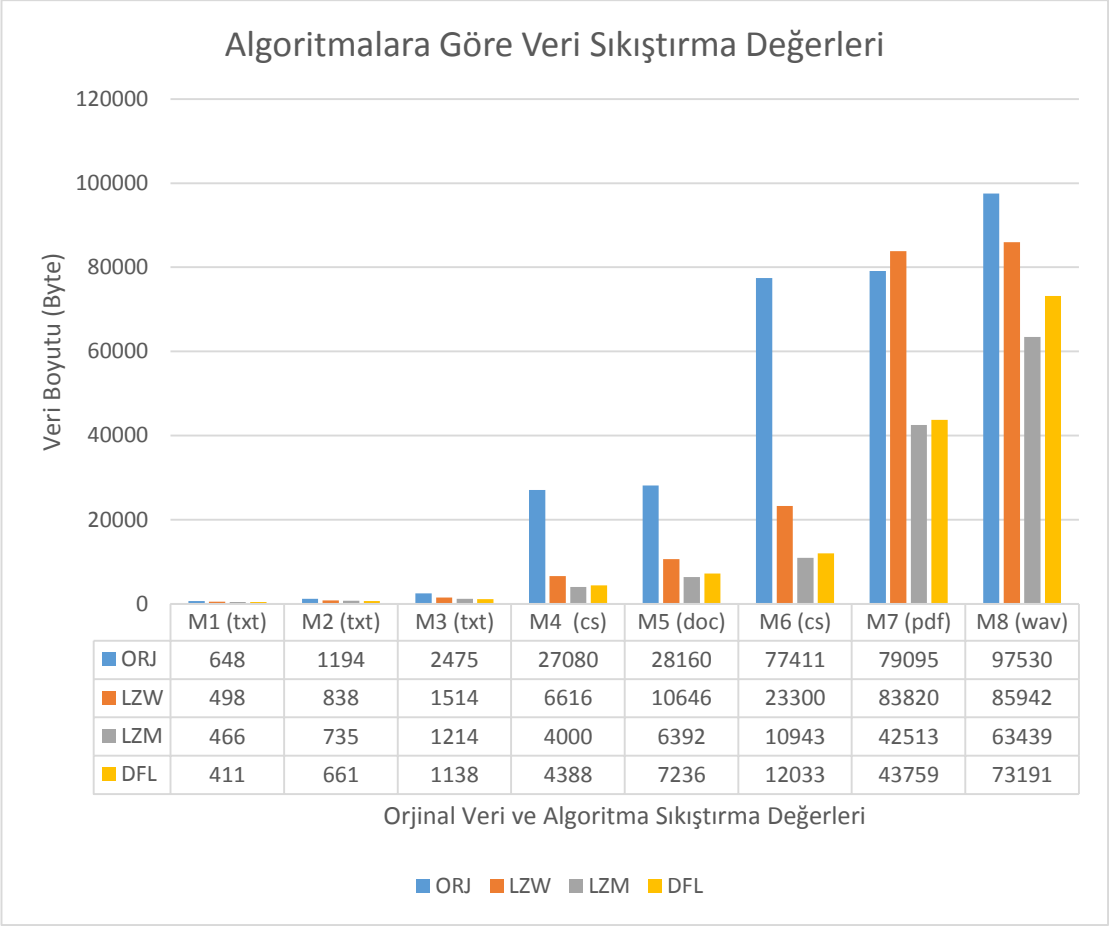
Gizlenmek istenilen veri kullanıcıdan alındıktan sonra ilk aşama LZW algoritmasını kullanarak yeni bir sözlük oluşturmaktır. Bu algoritmanın adımları ve çalışma mantığı üst başlıklarda örnek verilerek anlatılmıştı. Sözlük oluşturma ve metin sıkıştırma işlemlerinin gerçekleştirildiği uygulama kod örneği Şekil 4.12.'de verilmiştir.

```
public  Dictionary<string, int> KaynakSozluk;
public  List<int> sozlukdizi;

public  Dictionary<string, int> KaynakSozlukOlustur()
{    // Sıkıştırma işlemi için Sözlük Oluştur
   Encoding enTr = Encoding.GetEncoding("windows-1254"); // (1254)   TÜRKÇE KARAKTER
       KODLAMASI İÇİN

   byte[] unicodeBytes = new byte[256]; //unicode.GetBytes(dataTextBox.Text);

   KaynakSozluk = new Dictionary<string, int>();
   for (int a = 0; a <= 255; i++)
      unicodeBytes[a] = (byte)i;

   for (int a = 0; a <= 255; i++)
     KaynakSozluk.Add(enTr.GetString(unicodeBytes, a, 1), a);//a. anahtara
   return KaynakSozluk;
}

public  List<int> DataCompress(string orginalText)
{    // Sıkıştırma işlemi için Sözlük Oluştur (ilk 255 karakter)
   KaynakSozlukOlustur(); // :: KaynakSozluk
   string x = string.Empty;
   sozlukdizi = new List<int>();
   foreach (char y in orginalText) // 2 li, 3 lü, n li karşılaştırma
   {
      string xy = x + y;
      if (KaynakSozluk.ContainsKey(xy)) // yeni oluşan karakter (xy) sözlükte varmı yokmu?
         x = xy;
      else
      {
         sozlukdizi.Add(KaynakSozluk[x]);
         KaynakSozluk.Add(wc, KaynakSozluk.Count); // yeni sözcük-kelime için sözlüğe ekleme yap
         x = y.ToString();
      }
   }

   if (!string.IsNullOrEmpty(x))
      compressed.Add(KaynakSozluk[x]);
   return sozlukdizi;
}
```

Şekil 4.12. Sözlük oluşturma ve veri sıkıştırma C# kod parçası

Kullanıcıdan alına orijinal metin “DataCompress” metodu ile sıkıştırma işlemine tabi tutulur. Metodun başında yer alan “KaynakSozlukOlustur” fonksiyonu sıkıştırma işlemi için 0-255 byte değerleri arasındaki byte türündeki sayıların ASCII değerlerini alarak KaynakSozluk isminde ve Dictionary türündeki sözlüğe sırasıyla ekler. Böylece genişletilmiş ASCII olarak daha önceden anlatılan temel sözlük oluşturulmuş olur. Kodun devamında orijinal metin karakterleri sırasıyla tekli, ikili, üçlü, …, n’li olarak temel sözlükte aranır, mesaj verisinden alınan karakter daha önce sözlükte var ise bu karakterin yanına mesaj verisinin bir sonraki karakteri ilave edilir

ve sözlüğe yeni ananhtar metin olarak eklenir. Bu ananhtar metne karşılık gelen sayısal kodlama değeri ise sözlüğün o andaki toplam eleman sayısının bir fazlası olarak atanır ve bu değerde yeni anahtara karşılık gelen sayısal kodlama olarak belirlenir. Elde edilen yeni anahtar metin değeriyle daha sonraki tarama işlemlerinde karşılaşılması halinde, bu anahtar kelimeye mesaj verisinin bir sonraki karakteri ilave edilir ve böylece birden çok karakterden oluşan n uzunluğundaki söz/sözcük öbekleri tek bir sayısal değer ile ifade edilebilmiş ve kodlanmış olur. Uygulama içerisinde kullanılan farklı dosya türlerine ait mesaj verilerinin sıkıştırma algoritmaları sonucunda elde edilen sıkıştırımış veri boyutları Şekil 4.13.'de verilmiştir.



	M1 (txt)	M2 (txt)	M3 (txt)	M4 (cs)	M5 (doc)	M6 (cs)	M7 (pdf)	M8 (wav)
ORJ	648	1194	2475	27080	28160	77411	79095	97530
LZW	498	838	1514	6616	10646	23300	83820	85942
LZM	466	735	1214	4000	6392	10943	42513	63439
DFL	411	661	1138	4388	7236	12033	43759	73191

Şekil 4.13. Orjinal mesaj boyutları ve sıkıştırma sonrası elde edilen yeni boyutlar

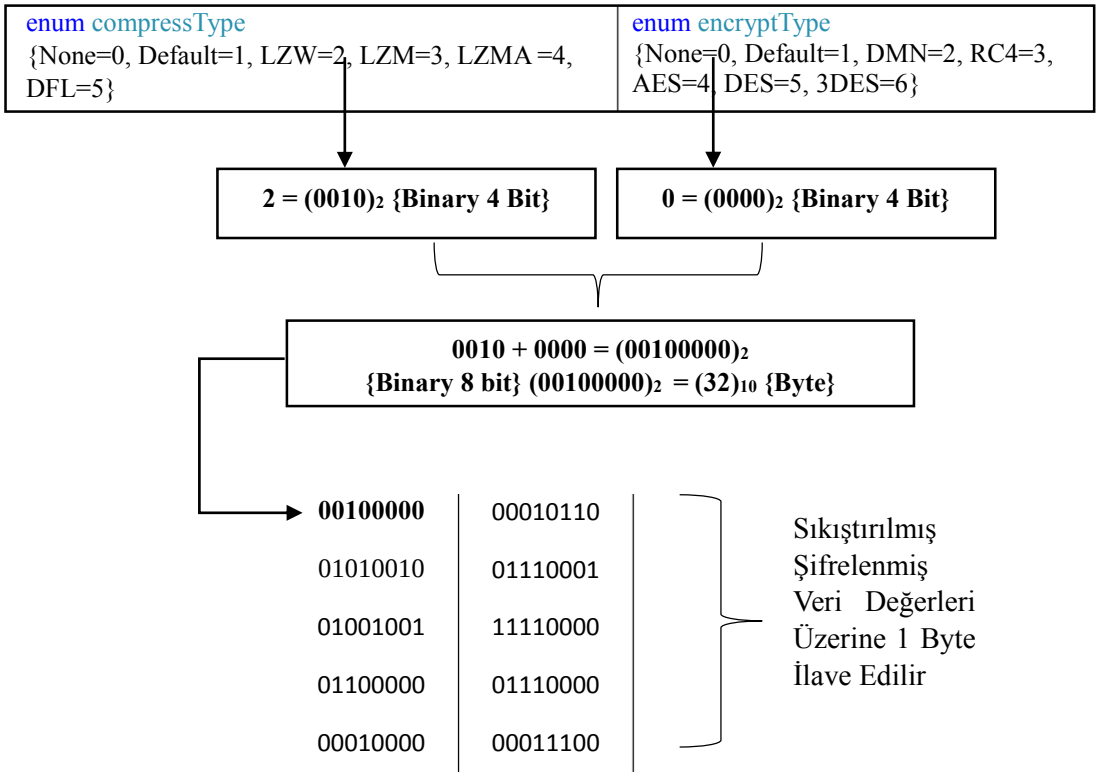
Elde edilen ve Şekil 4.13.'de gösterilen değerlere bakıldığında dosya tiplerine göre uygulanan sıkıştırma algoritmalarının farklı sonuçlar verdiği gözlenmektedir. Bu algoritmalar içerisinde 7Zip algoritmasının da temelini oluşturan LZM

algortmasından türetilmiş olan sıkıştırma algoritması en iyi değerleri sunan sıkıştırma tekniği olarak görülmüş ve uygulama içerisinde varsayılan sıkıştırma tekniği olarak kullanılmıştır.

Daha sonraki adımda ise yeni sözlüğe göre elde edilen ve sıkıştırılmış olan gizli bilgi, kullanıcı tarafından girilen anahtar değeriyle seçeceği simetrik blok şifreleme(AES, DES, 3DES) veya simetrik dizi şifreleme (RC4, DMN) algoritmalarından herhangi birisi ile kodlanarak yeniden üretilebilir. Sıkıştırma algoritması ve simetrik şifreleme işlemi uygulanan mesaj verisi taşıyıcı dosya içerisine gömülmeye hazır hale gelmiştir.

Veri sıkıştırma ve şifreleme işlemi uygulanmadan da orijinal mesaj verisi ham haliyle taşıyıcı dosya içerisine gizlenebilmektedir. Bunun tam aksine daha önceki konu başlığında ifade edildiği gibi orijinal mesaj/dosya herhangi bir taşıyıcı içerisine gizlenmeden sadece veri sıkıştırma, sadece veri şifreleme veya bunların her ikisininde uygulandığı bir kombinasyon işlemlerine tabi tutulduktan sonra elde edilen yeni yapı çıktı olarak kaydedilebilir.

Orijinal veri üzerinde gizleme işlemi dışında elde edilen yeni yapıya hangi işlemlerin (sıkıştırma, şifreleme) uygulanıp uygulanmadığı bilgisini saklayabilmek için elde edilen yeni yapı üzerine bir (1) byte'lık veri ilave edilmektedir. Bu bir byte'lık verinin ilk 4 byte değeri sıkıştırma değerini tutarken, son 4 byte değeri ise şifreleme yapılıp yapılmadığını yapıldıysa hangi algoritmanın uygulandığı değerini saklamaktadır. Şekil 4.14.'de bu veri ilave işlemi gösterilmiştir.



Şekil 4.14. Kodlanmış mesaj verisinin üzerine kontrol bitlerinin eklenmesi

11010000 00010001

Kodlanmış mesaj seçilecek taşıyıcı bir ses dosyasının en az öneme sahip bitleri (Least Significant Bit) üzerine, girilen anahtar değerine göre rastgele (karmaşık düzende) yerleştirilmektedir. Gizleme işlemini yapmadan önce seçilen taşıyıcı dosyanın uygun formatta olup olmadığına bakılması gerekiyor. Daha önceki üst başlıklarda anlatıldığı gibi uygun bir ses (wav) dosyasının ilk 4 byte'i "RIFF" değerini, sonraki 4 byte "Bölüm Boyutunu" bilgisini, sonraki 4 byte "WAVE" değerini, sonraki 4 byte "fmt " değerini, sonraki 4 byte "Alt Bölüm Boyutunu" bilgisini içermelidir. Ses dosyasının 36-40 byte'ları ise "data" verisini içermelidir. Bu değerlerin okunduğu ve bu uygulamda da kullanılan örnek bir ses dosyasının ilk 44 ve 72 byte lık bloklara ait değerler sırasıyla Tablo 4.4. ve Tablo 4.5.'de gösterilmiştir.

Tablo 4.4. Örnek bir ses dosyasının ilk 44 byte'lık verileri ve değerleri

Kaynak Pozisyon ========== [Byte] 0 00000 : 01010010 : 082: R 00001 : 01001001 : 073: I 00002 : 01000110 : 070: F 00003 : 01000110 : 070: F Kaynak Pozisyon ========== [Byte] 4 ChunkSize | Bölüm Boyutu: 1823632 Kaynak Pozisyon ========== [Byte] 8 00008 : 01010111 : 087: W 00009 : 01000001 : 065: A 00010 : 01010110 : 086: V 00011 : 01000101 : 069: E Kaynak Pozisyon ========== [Byte] 12 00012 : 01100110 : 102: f 00013 : 01101101 : 109: m 00014 : 01110100 : 116: t 00015 : 00100000 : 032: Kaynak Pozisyon ========== [Byte] 16 Subchunk1 Size | Alt Bölüm Boyutu: 16 Byte Kaynak Pozisyon ========== [Byte] 20 Audio Format | Ses Biçimi: 1	Kaynak Pozisyon ========== [Byte] 22 Num Channels | Kanal Sayısı: 1 Kaynak Pozisyon ========== [Byte] 24 Sample Rate | Ses Oranı: 44100 Hz Kaynak Pozisyon ========== [Byte] 28 SampleRate x NumChannels x BitsPerSample/8 Byte Rate | Bayt Oranı: 88200 Byte Kaynak Pozisyon ========== [Byte] 32 Block Align | Blok Sırası: 2 Kaynak Pozisyon ========== [Byte] 34 Bits Per Sample | Bir Ses Örn. Bit Sayısı: 16 Bit Kaynak Pozisyon ========== [Byte] 36 00036 : 01100100 : 100: d 00037 : 01100001 : 097: a 00038 : 01110100 : 116: t 00039 : 01100001 : 097: a Kaynak Pozisyon ========== [Byte] 40 NumSamples x NumChannels x BitsPerSample/8 Subchunk2 Size | Alt Bölüm2 (Ses Verilerinin) Boyutu: 1823514 Byte Kaynak Pozisyon ========== [Byte] 44

Tablo 4.5. Örnek bir ses dosyasının ilk 72 byte'lık verileri

BYTE BINARY Hex Char	**BYTE BINARY Hex Char**	**BYTE BINARY Hex Char**
Kaynak Pozisyon ====== 0 00000 : 01010010 : 082: R 00001 : 01001001 : 073: I 00002 : 01000110 : 070: F 00003 : 01000110 : 070: F 00004 : 10010000 : 144: □ 00005 : 11010011 : 211: Ó 00006 : 00011011 : 027: _ 00007 : 00000000 : 000: nul 00008 : 01010111 : 087: W 00009 : 01000001 : 065: A 00010 : 01010110 : 086: V 00011 : 01000101 : 069: E 00012 : 01100110 : 102: f 00013 : 01101101 : 109: m 00014 : 01110100 : 116: t 00015 : 00100000 : 032: 00016 : 00010000 : 016: _ 00017 : 00000000 : 000: nul 00018 : 00000000 : 000: nul 00019 : 00000000 : 000: nul 00020 : 00000001 : 001: _ 00021 : 00000000 : 000: nul 00022 : 00000001 : 001: _	00023 : 00000000 : 000: nul 00024 : 01000100 : 068: D 00025 : 10101100 : 172: ¬ 00026 : 00000000 : 000: nul 00027 : 00000000 : 000: nul 00028 : 10001000 : 136: □ 00029 : 01011000 : 088: X 00030 : 00000001 : 001: _ 00031 : 00000000 : 000: nul 00032 : 00000010 : 002: _ 00033 : 00000000 : 000: nul 00034 : 00010000 : 016: _ 00035 : 00000000 : 000: nul 00036 : 01100100 : 100: d 00037 : 01100001 : 097: a 00038 : 01110100 : 116: t 00039 : 01100001 : 097: a 00040 : 00011010 : 026: _ 00041 : 11010011 : 211: Ó 00042 : 00011011 : 027: _ 00043 : 00000000 : 000: nul 00044 : 01000000 : 064: @ 00045 : 00000000 : 000: nul 00046 : 01100000 : 096: ` 00047 : 00000000 : 000: nul	00048 : 10100000 : 160: 00049 : 00000000 : 000: nul 00050 : 01110000 : 112: p 00051 : 00000000 : 000: nul 00052 : 00100000 : 032: 00053 : 00000000 : 000: nul 00054 : 10000000 : 128: □ 00055 : 00000000 : 000: nul 00056 : 00010000 : 016: _ 00057 : 00000001 : 001: _ 00058 : 11110000 : 240: ð 00059 : 00000000 : 000: nul 00060 : 10000000 : 128: □ 00061 : 00000000 : 000: nul 00062 : 01100000 : 096: ` 00063 : 00000000 : 000: nul 00064 : 10010000 : 144: □ 00065 : 00000000 : 000: nul 00066 : 10110000 : 176: ° 00067 : 00000000 : 000: nul 00068 : 10100000 : 160: 00069 : 00000000 : 000: nul 00070 : 00000000 : 000: nul 00071 : 00000001 : 001: _ Kaynak Pozisyon ====== 72

Doğru bir formata sahip olan ses (wav) dosyasının içinde asıl ses verileri 44 uncü byte'dan sonra başlamaktadır. Ses dosyasının başlık bilgilerini okuyan ve dosya yapısını kontrol etmemizi sağlayan algoritma adımları aşağıda verilmiştir. Algoritmada kullanılan K: Taşıyıcı ses dosyasının (Kaynak) binary formatını ifade etmektedir.

Dosya yapısı kontrol adımları:

- Başla
- Set K.Pozisyon ← 0 // Kaynak ses dosyasının 0 ıncı byte ına git.
- IF (**ByteOku**(K, 4) != ″RIFF″) : Exit // *Kaynaktan 4 byte oku ve ASCII değerine bak. Okunan adet (byte) kadar Kaynak dosya üzerinde ilerle*
- Set Bölüm_Boyutu = K.ReadIn32() // *Kaynaktan 4 byte lık veri oku ve ilerle*
- IF (**ByteOku**(K, 4) != ″WAVE″) : Exit
- IF (**ByteOku**(K, 4) != ″fmt ″) : Exit
- Set Alt_Bolum_Boyutu ← K.ReadIn32() < 16: Exit *// 4 byte oku*
- Set Ses_formatı ← K.Read16() *// 2 byte oku*
- Set Kanal_sayısı ← K.Read16()
- Set Ses_Oranı ← K.Read32()
- Set Byte_Oranı ← K.Read32()
- Set Blok_Sırası ← K.Read16()
- Set Ses_Ornegi_Bit_Sayısı ← K.Read16()
- IF (**ByteOku**(K, 4) != ″data ″) : Exit
- Set Ses_Verilerinin_Boyutu ← K.Read32()
- Bitir

Algoritma içerisinde verilen ve geriye string türünde değer döndüren ByteOku fonksiyonuna ait işlem adımları aşağıda verilmiştir;

ByteOku(*BinaryReader reader**, int** adet***)**: Return **String**

- Başla
- Set dizi ← new Byte[adet] //*adet uzunluğunda dizi oluştur*

- reader.Read(dizi, 0, adet) // *reader nesnesinde adet kadar byte değeri oku ve dizi elemanına 0 ıncı indexten itibaren yaz.*
- Out ← Encoding.ASCII.GetString(dizi) *// byte olarak değer saklayan dizinin herbir elemanının ASCII karakter karşılığını al*.
- Return ← Out
- Bitir

Gizlenecek verinin boytu, taşıyıcı nesnenin LSB bitlerinin boyutunu aşmamalıdır. Bu durumda veriyi gömmek için LSB bitleri dışında kalan bitlere de yerleştirebiliriz. Dikkat edilmesi gereken husus stego nesnede meydana gelebilecek bozulmalar algılanamazlık kriterini olumsuz yönde etkileyeceğidir. Ses dosyasının sır açma (stegaanaliz) yöntemleri ile irdelenmesi sonucu, LSB bitlerine erişilmesi halinde, ses dosyası içerisine gömülü olan verilere ulaşılmamalıdır. Bu sebeple verileri taşıyıcı nesnenin LSB bitlerine sıralı olarak değil karmaşık bir düzende gömmek daha etkili bir yöntem olacaktır. Bunu yapmamızı sağlayacak olan da kullanacağımız gömme algoritmasına bağlıdır. Geliştirilen uygulamda kullanılan veri gizleme algoritma adımları aşağıda verilmiştir.

Algoritma içerisinde kullanılan kısaltmalar;

M: Mesaj verisi binary stream	M_i: Mesajın i ninci değeri (byte)
K: Anahtar verisi binary stream	K_i: Anahtarın i ninci değeri (byte)
n: Mesaj verisinin uzuluğu	m: Anahtar verisinin uzunluğu
S: Source/Kaynak ses dosyası	D: Destination/Hedef ses dosyası

Veri gömme algoritma adımları:

- Başla
- M.Pozisyon = 0
- K.Pozisyon = 0
- waveBuffer = new byte[bytesPerSample]
- While (Mi = ReadByte ($M = \sum_0^n M_0 + \cdots + M_n$) ≥ 0)

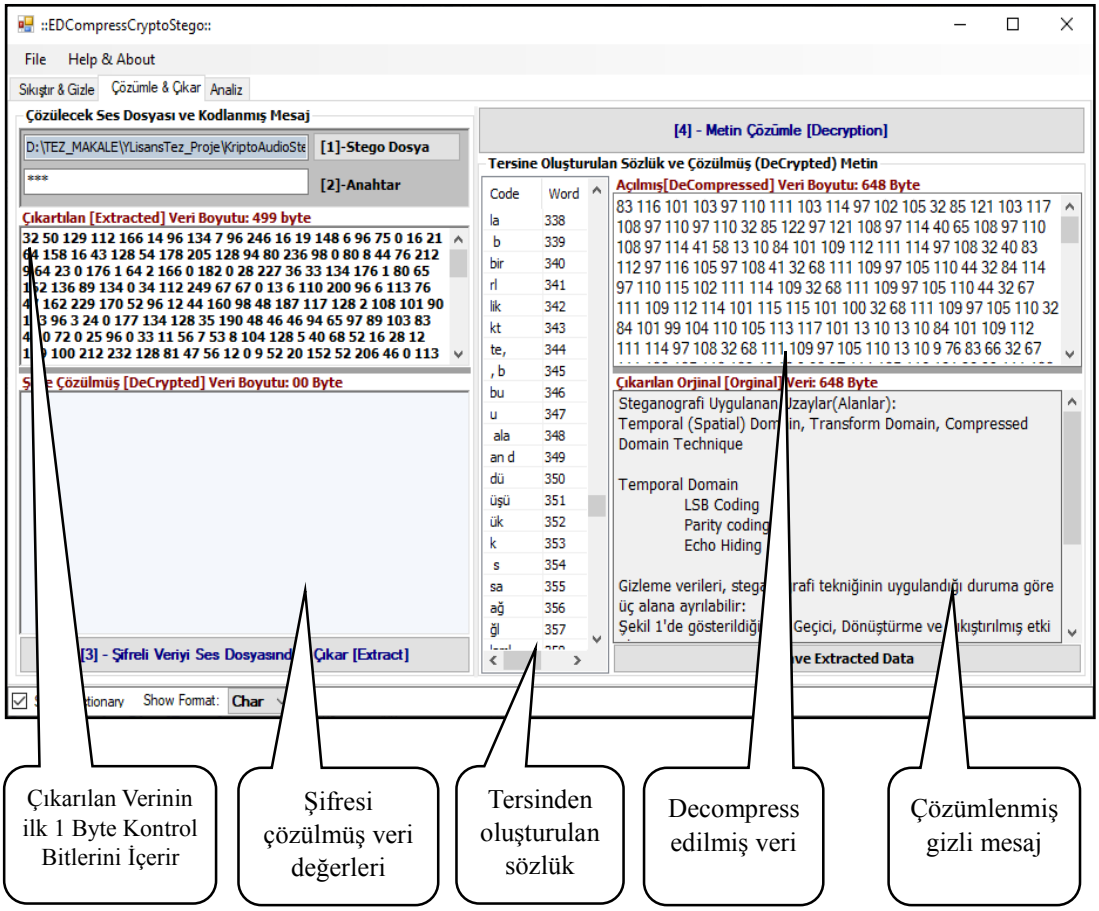
1. Set message ← (byte)M_i
2. For bitindex = 0 to 8
   - Ki $= \begin{cases} K_i, & i < m \\ K_0, & i \geq m \end{cases}$
   - sayac = $(int)K_i/8$ // *Değiştirilmeden atlanacak bit sayısı*
   - For j = 0 to sayac -1 *//sayac kadar bit değiştrilmeden kaynaktan(S) hedefe(D) kopyala*
     S.Copy (waveBuffer, 0, waveBuffer.length, D)
   - S.Read (waveBuffer, 0, waveBuffer.length)
   - waveByte = waveBuffer[bytesPerSample - 1]
   - islem = (byte)(1 << bitIndex)
   - islemsonuc = (byte)(message & islem)
   - bit = (byte)((islemsonuc > 0) ? 1 : 0)
   - if ((bit == 1) && ((waveByte % 2) == 0))
     waveByte += 1
   - else if ((bit == 0) && ((waveByte % 2) == 1))
     waveByte -= 1
   - waveBuffer[bytesPerSample - 1] = waveByte
   - D.Write(waveBuffer, 0, bytesPerSample)

- End While //*Mesaj verisi byte byte okundu ve karmaşık bir düzen içerisinde hedef dosyaya yazıldı*
- waveBuffer = new byte[S.Length - S.Position] //*Kaynaktan kalan veri uzunluğu kadar diziyi tanımla*
- S.Read(waveBuffer, 0, waveBuffer.Length) // *Kalan veri uzunluğu kadar* k*aynak dosyadan veri oku ve diziye at*
- D.Write(waveBuffer, 0, waveBuffer.Length) // *Diziden al hedef dosyaya yaz*
- Bitir.

4.5.2. Ses dosyasından veri çıkarma ve kod çözme

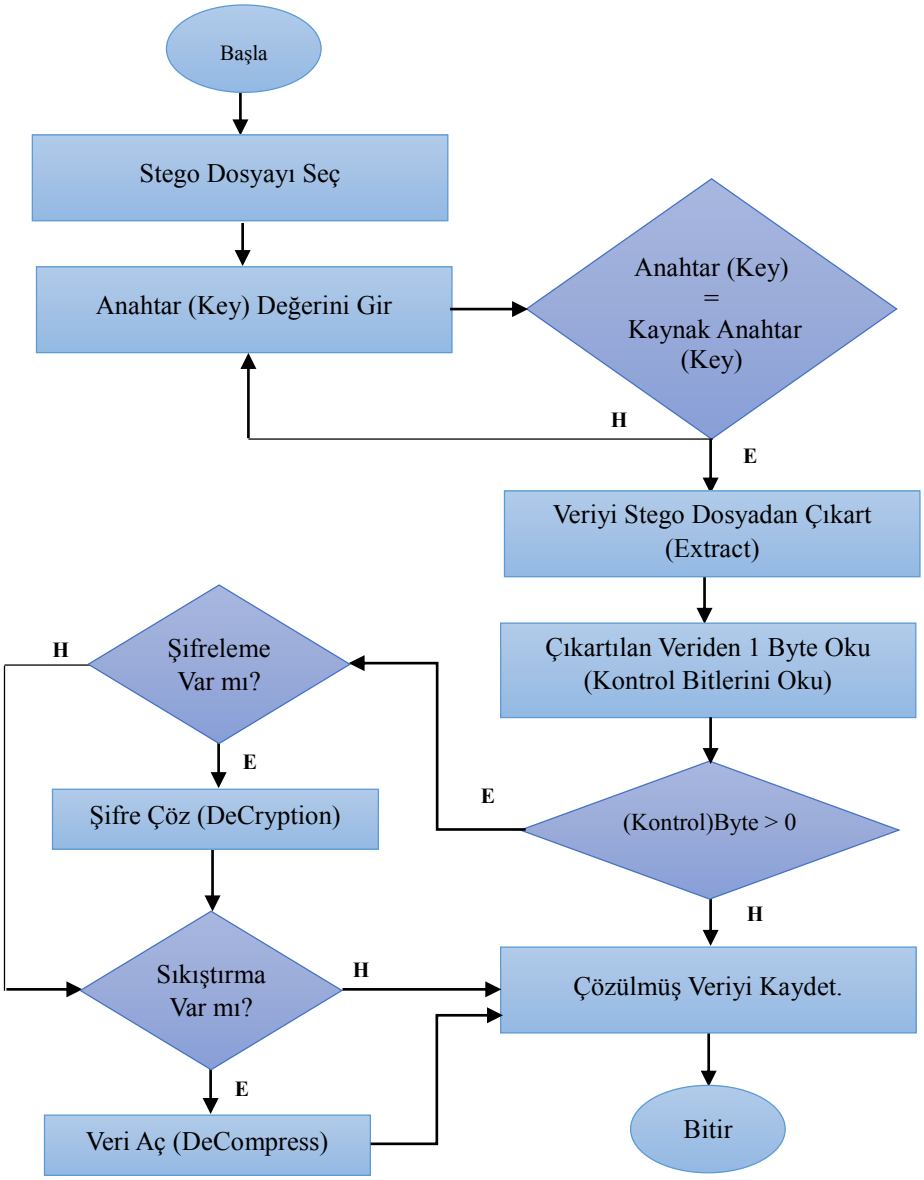
Uygulamanın bu bölümü işleyiş bakımından temel olarak üç adımdan oluşmaktadır. Birinci adım ses dosyası içerisine gömülü olan mesajın çıkartılmasını (extract)

sağlarken, ikinci adım dosyadan çıkarılan veri gizlenmeden önce şifrelenmiş ise kullanılan şifreleme algoritmasına göre şifre çözme (decryption) işlemi gerçekleştirilir. Üçüncü adımda ise gizli veri üzerine herhangi bir sıkıştırma işlemi uygulanmış ise sıkıştırılmış veriye ait byte değerleri kullanılarak tersinden sözlük oluşturulup veri açma/çözümleme (decompress) işlemi gerçekleştirilmektedir.



Şekil 4.15. Uygulama veri çıkarma ve kod çözme ara yüzü

Veri çıkarma (extract), şifre çözme (decryption) ve veri açma (decompress) işlemlerinin gerçekleştirildiği uygulama arayüzü Şekil 4.15.'de, bu işlem adımlarının gerçekleşme sırasını gösteren iş akış adımları Şekil 4.16.'da gösterilmiştir.



Şekil 4.16. Veri çıkartma, şifre çözme, veri açma işlem adımları

Gizli veri üzerinde saklama işlemi yapılmadan önce hangi işlemlerin (sıkıştırma, şifreleme her ikisi, hiç biri) uygulanıp uygulanmadığı bilgisine bakılarak mesaj çözümleme adımları gerçekleştirilmektedir. Buna göre stego dosyadan çıkartılan veriden bir byte'lık değer alınır ve onluk tabanda karşılığına bakılır. Eğer bu değer sıfır'dan (0) büyük ise bu verilere gizleme işleminden önce sıkıştırma, şifreleme işlemlerinden biri veya her ikisi uygulanmış demektir. Bu aşamada elde edilen 1 byte'lık veri ikli (binary) formatta yazılır ve dörder bit şeklinde iki parça blok olarak yeniden değerlendirmeye tabi tutulur. Bu blokların ilk dört biti sıkıştırma yapılıp yapılmadığını son dört biti ise şifreleme yapılıp yapılmadığı bilgisini saklar. Bu

kontrol bitlerinin nasıl oluşturlduğu bir önceki konu başlığında Şekil 4.14'de detaylı bir şekilde anlatılmıştı. Ses dosyasından veriyi çıkartan uygulama kod parçası Şekil 4.17.'de verilmiştir.

```
public void Cikart(Stream mesajDizi, Stream anahtarDizi)
{
   keyDizi.Position = 0;

   byte[] sesTampon = new byte[ornek];
   byte mesaj, bit, sesByte;
   int mesajUzunluk = 0;
   int anahtarByte;
   int sayac;
   while ((mesajUzunluk == 0 || mesajDizi.Length < mesajUzunluk))
   {
      mesaj = 0;  //mesaj değerini sıfırla

      for (int bitIndex = 0; bitIndex < 8; bitIndex++) //mesajdan 1 bit al
      {
         anahtarByte = GetKeyValue(anahtarDizi); //anahtardan 1 bayt al
         sayac = (int)( anahtarByte / 8);  //değiştirilmeden kopyalanacak olan değerler
         for (int n=0; n< sayac; n++)
         {
            //kaynak dosyadan belirlenen değer kadar okuma yap
            kaynakSesDosya.Read(sesTampon, 0, sesTampon.Length);
         }
         sesByte = sesTampon [bytesPerSample - 1];
         bit = (byte)(((sesByte % 2) == 0) ? 0 : 1); //ses örneğinin son bitini al...
         mesaj += (byte)(bit << bitIndex); //...mesaj değerini yaz
      }

      //elde edilen mesaj değerini diziye ekle
      mesajDizi.WriteByte(mesaj);

      if (mesajUzunluk == 0 && mesajDizi.Length == 4)
      {
         //İlk 4 byte mesajın uzunluğunu içerir
         mesajDizi.Seek(0, SeekOrigin.Begin);
         mesajUzunluk = new BinaryReader(mesajDizi).ReadInt32();
         mesajDizi.Seek(0, SeekOrigin.Begin);
         mesajDizi.SetLength(0);
      }
   } // End While
} // End Cikart
```

Şekil 4.17. Stego dosyadan veri çıkaran uygulama kod parçası

Gönderilen gizli mesajın ses dosyası içerisinden çıkarılması ve kodlanmış bilginin çözümlenmesi ise bir önceki konu başlığında anlatılan ve algoritma adımları verilen işlemlerin tersi adımlarla gerçekleştirilmesi sonucu elde edilmektedir.

Şekil 4.15.'de görüldüğü gibi gizli bilginin saklandığı stego dosya ve anahtar değeri kullanıcı tarafından girilmektedir. Taşıyıcı dosya içerisine gizlenen veriler, kullanıcı anahtar değerine bağlı olarak ilgili ses örneklerinin son bitlerine yerleştirilmişti. Girilen anahtar değerine bakılarak veri gömme algoritması benzer şekilde tersden işletilerek gizli veriler ses dosyasının ilgili ses örneklerinin son bitlerinden okunarak geri elde edilir. Çıkarılan veriler gönderici tarafından bağımsız olarak alıcı tarafta yeniden sözlük oluşturma adımlarını gerçekleştirmektedir.

```
    public string DataDecompress(List<int> compressed)
    {
        //  Veri Açma işlemi için sözlük oluştur
        Encoding enTr = Encoding.GetEncoding("windows-1254"); // (1254)
        byte[] unicodeBytes = new byte[256]; //unicode.GetBytes(dataTextBox.Text);

        HedefSozluk = new Dictionary<int, string>();
        for (int j = 0; j <= 255; i++)
            unicodeBytes[j] = (byte)i;

        for (int j = 0; j <= 255; i++)
            HedefSozluk.Add(j, enTr.GetString(unicodeBytes, j, 1));

        string w = HedefSozluk[compressed[0]];
        sozlukDizi.RemoveAt(0);
        StringBuilder tersineSozluk = new StringBuilder(w);
        foreach (int m in sozlukDizi)
        {
            string giris = null;
            if (HedefSozluk.ContainsKey(m))
                giris = HedefSozluk[m];
            else if (m == HedefSozluk.Count)
                giris = x + x[0];

            tersineSozluk.Append(giris);
            // yeni dizi sözlüğe ekleniyor
            HedefSozluk.Add(HedefSozluk.Count, x + giris[0]);
            x = giris;
        }
        return tersineSozluk.ToString();
    }
}
```

Şekil 4.18. Tersinden sözlük oluşturan ve gizli mesajı stego dosyadan çıkaran kod parçası

Stego dosyadan veri çıkartıldıktan ve deşifreleme işlemi uygulandıktan sonra, veri sıkıştırmada oluşturulan LZW sözlük tersinden oluşturulur ve LZW algoritmasına göre kodlanmış olan veri çözümlenerek orijinal mesaj yeniden elde edilir. Gizli mesajı çözümleyen uygulama kod parçası Şekil 4.15.'de verilmiştir.

4.6. EDStego yazılımı ve diğer stego yazılımların karşılaştırlması

Günümüzde veri güvenliği ve gizliliği adına farklı dillerde ve farklı işletim sistemi platformları için geliştirilmiş olan birçok steganografi yazılımı mevcuttur. Bu yazılımların bir kısmı ücretli olmakla beraber ücretsiz olarak kullanıma sunulan uygulamalarda mevcuttur. Geliştirilen steganografik yazılımlarının çoğu taşıyıcı nesne olarak resim dosyalarını kullanmaktadır. Ses dosyaları içerisine veri gizleyen yazılım sayısı diğer stego yazılımlarına göre oldukça azdır. EDstego uygulaması hem resim (bmp, png, ico) hemde ses dosyaları (wav) içerisine veri gizleyen ender stego uygulamalarından birisidir. Tablo 4.6.'da bu stego yazılımlarının bünyesinde barındırdıkları taşıyıcı nesne, saklanan veri türü, şifreleme kullanıp kullanmadığı, veri sıkıştırma yapıp yapmadığı gibi temel özellikler üzerinde karşılarştırma değerleri verilmiştir. Bu uygulamalar içerisinde ses dosyaları içerisine veri gizleyebilen yazılımlar kalın olarak gösterilmiştir. EDstego yazılımı gizlenecek veri/dosya üzerine kayıpsız veri sıkıştırma tekniklerini kullanan az sayıdaki stego uygulamalarıdan biri olarak görülmektedir. Uygulama arayüz dili olarak Türkçe dilinde yazılmış olan birkaç yazılımdan biridir.

Tablo içerisinde kullanılan;

- R: Resim dosyalarını
- S: Ses dosyalarını
- V: Video dosyalarını
- Doc: Çeşitli (dökuman) uygulama dosyalarını
- D: Diğer alanları temsil etmektedir

Tablo 4.6. EDStego yazılımı ile diğer stego yazılımlarının karşılaştırılması

Uygulama	Uygulama Özellikleri					
	Kullanılan Taşıyıcı Nesne	**Dosya Türü**	**Şifreleme**	**Sıkıştırma**	**Dil**	**Açıklama**
BMPSecrets	**R:** gif, tiff, jpg, bmp	Metin	-	-	İngilizce	
DarkCryptTC	**R:** gif, tiff, jpg, png, sd, tga, tng **S:** wav **Doc**: odt, html, xml, txt **D:** exe, dll, ntfs katarları	Hepsi	Var	-	İngilizce	RSD modu (RNG tabanlı rasgele veri dağılımı) kullanılmıştır. AES şifreleme desteklenir.
DeepSound	**R:** bmp **S:** wav, mp3, audio cd, ape tag, flac, wma	Hepsi	Var	-	İngilizce	256 bit AES şifrelemeyi kullanır
EDStego	**R:** bmp, png, ico **S:** wav	Hepsi	Var	Var	Türkçe	Açık Kaynak LSB tekniğini kullanır. AES, DES, 3DES, RC4 şifrelemeyi kullanır. LZW, LZM, LZMA, 7ZIP veri sıkıştırmayı kullanır. Sadece sıkıştırma ve şişfreleme içinde kullanılabilir.
ImageSpyer G2	**R:** tiff, bmp	Metin	-	-	İngilizce	RSD (Reciprocal smallest distance algorithm - Karşılıklı en küçük mesafe) algoritmasını kullanır
Hermetic	**R:** bmp, jpg, png, pcx, tga, aiff **S:** Wav, mp3	Hepsi	Var	-	İngilizce	
MP3Stego	**S:** mp3	Metin	Var	-	İngilizce	Açık kaynak
Mr. Crypto	**R:** tiff, png, bmp	Metin	Var	-	İngilizce	AES, 3DES şifrelem algoritmalarını kullanır. LSB tekniğine göre gizleme yapar
OpenPuff	**R:** tga, png, jpeg, bmp **S:** wav, mp3 **V:** mpeg-2, mp4, mpeg-1, 3gp, vob, flv, swf,	Hepsi	Var	-	İngilizce İtalyanca	Birden fazla taşıyıcı dosyaya veri gizleyebilir. Kriptografi, steganografi ve beyazlatma tekniklerini içinde barıdırır. Klasör gizleme özelliği vardır.
OpenStego	**R:** bmp, png				İngilizce	Açık kaynak
Outguess	**R:** jpg	Metin	-	-	İngilizce	
QuickStego	**R:** bmp, jpeg, gif	Metin	-	-	İngilizce	
S-Tools	**R:** bmp, gıf **S:** wav **D:** Kullanılmayan disk alanları	Hepsi	-	-	İngilizce	

Tablo 4.6. (Devamı)

Steg	**R:** bmp, png, jpeg, gif	Hepsi	Var	-	İngilizce	Simetrik ve asimetrik şifreleme kullanır. Win/Linux/Mac de çalışır.
StegaMail	**R:** bmp, png	Metin	Var	Var	İngilizce	56 bit şifreleme yapar. zLib sıkıştırma kullanır.
Steganographic Laboratory (VSL)	**R:** tiff, png, jpg, bmp	Hepsi	-	-	İngilizce	Açık kaynak Bütün platformlarda çalışabilir
Steganography Studio	**R:** gıf, bmp, png	Hepsi	-	-	İngilizce	Açık kaynak Farklı gizleme tekniklerini (LSB, LSB Matching, SLSB) kullanılır. İçerisinde resim steganaliz tool barındırır.
StegHide	**R:** jpg, bmp **S:** wav, au	Hepsi	Var	-	İngilizce	Açık kaynak
StegoShare	**R:** bmp, jpeg, png, gif, tiff				İngilizce	Açık kaynak
STTK	**S:** wav	Hepsi	Var	-	Türkçe	Birden fazla dosya içerisine veri gizleme yapabilir. AES şifrelemeyi destekler. LSB tekniğini kullanır. Klasör gizleme özelliği vardır.
TurkSteg	**R:** bmp	Metin	Var	-	Türkçe	

BÖLÜM 5. UYGULAMA ANALİZLERİ VE SONUÇ

Bu bölümde geliştirilen uygulama içinde kullanılan teknikler için performans ölçümleri yapılmıştır. Geliştirilen uygulamanın performansını etkileyen ölçütler kapasite, güvenlik, algılanamazlık olarak nitelendirilebilir. Bu bağlamda gerçekleştirilen veri sıkıştırma işlemi kapasite analizinin temelini oluştururken, ses dosyalarının içerisine saklanan veri ve buna bağlı olarak taşıyıcı dosyada meydana gelen bozulma oranları ise algılanamazlık analizini teşkil etmektedir. Steganografi uygulamalarında güvenlik analizi ise gizli mesaja ulaşılamamasını veya ulaşılsa bile çözülememesini temel almaktadır.

5.1. Kapasite Analizi

Gerçekleştirilen bir strganografi uygulamalarında kapasite, gizlenecek mesajın boyutunun stego ortamın boyutuna oranı olarak nitelendirilmektedir [73]. Bu sebeple veri gömme işleminden önce gizlenecek mesajın sıkıştırma oranının başarısı büyük önem taşımaktadır.

$$C = \frac{Bits\_of\_Screet\_Message}{Bits\_Of\_Stego\_Cover}$$

LZW kaypsız veri sıkıştırma algoritması kullanılarak uygulama içersinde gizlenen verilerde değişik oranlarda sıkıştırma sağlanmıştır. Verilerdeki sıkıştırma oranları dosya türlerine, verinin uzunluğuna ve içerdiği tekrarlı sözcüklere göre değişiklik göstermektedir. Karakter sayısı arttıkça tekrar sayısı artacağından sıkıştırma performansı da artmaktadır. Yapılan çalışmada kullanılan metin/dosya örnekleri ve bu örnekler üzerinde yapılan sıkıştırma oranları Tablo 5.1.'de gösterilmiştir.

Sıkıştırma oranlarını hesaplamada üst başlıklarda anlatılan Denklem 4.3 ve Denklem 4.4'den faydalanılmıştır.

Tablo 5.1. Veri sıkıştırma kapasite analiz sonuçları

Orjinal Mesaj	Orijinal Dosya		Sıkıştırma Sonrası Veri Uzunluğu (Byte)		Sıkıştırma Oranı			
					LZW		LZM	
	Türü	Uzunluk (Byte)	LZW	LZM	C(1)	R(%)	C(1)	R(%)
M1	txt	648	498	466	1,3012	23,15	1,3906	28,09
M2	txt	1194	838	735	1,4248	29,82	1,6245	38,44
M3	txt	2475	1514	1214	1,6347	38,83	2,0387	50,95
M4	cs	27080	6616	4000	4,0931	75,57	6,77	85,23
M5	doc	28160	10646	6392	2,6451	62,19	4,4055	77,30
M6	cs	77411	23300	10943	3,3224	69,90	7,074	85,86
M7	pdf	79095	**83820**	42513	**0,9436**	**- 5,97**	1,8605	46,25
M8	wav	97530	85942	63439	1,1348	11,88	1,5374	34,95

Verilen örnek sonuçlarından da anlaşılacağı üzere yapılan bu sıkıştırma işlemi sonucunda, stego nesne üzerinde kapasite artılmış ve daha fazla bilgi gizleme imkânı oluşturulmuştur. Gizlenecek mesaj verisinde karakter sayısı arttıkça tekrar eden kelime sayısı da artacağından sıkıştırma performansının dahada artacağı rahatlıkla söylenilebilir.

Tablo 5.2. Uygulama işlem süreleri, taşıyıcı ve stego dosya boyutları

Gizlenenen Veri		Anahtar		İşlem Süreleri (sn)		Dosya Boyut Bilgileri (Byte)	
Mesaj	Boyut (Byte)	Key	Uzunluk	Sıkıştırma	Gizleme	Taşıyıcı	Stego
M1	648	ERD	3	0,0309731	0,0061753	1826816	1826816
M2	1194			0,0431128	0,0070138	1826816	1826816
M3	2475			0,0367021	0,0394199	1826816	1826816
M4	27080			0,1016696	0,0127031	1826816	1826816
M5	28160			0,0878791	0,0243886	1826816	1826816
M6	77411			0,2405946	0,0485724	1826816	1826816
M7	79095			0,1205898	0,103348	1826816	1826816
M8	97530			0,1577533	0,1106151	1826816	1826816

Ses dosyalarına yapılan veri gizleme işlemi sonrasında elde edilen stego ses dosyaları ile orijinal ses dosyalarının karşılaştırıldığında dosya boyutlarında hiçbir artışın olmadığı görülmektedir. Tablo 5.2.'de orijinal ses dosyası ve aynı anahtar kullanılarak bu ses dosyasına gizlenen farklı uzunlukta ve farklı dosya formatlarına sahip sekiz adet gizli mesaj verisinin gizleme sonrası dosya boyut bilgileri verilmiştir.

5.2. Algılanamazlık Analizleri

Gerçekleştirilen steganografik yapının algılanamazlık analizleri için SNR, PSNR ölçümleri, $\chi 2$ (ki-kare) testi, histogram analizleri ve spectrogram analizleri yapılmılmış ve bu analiz değerlerine ait sonuçlar alt başlıklar halinde anlatılmıştır.

SNR ve PSNR ölçüm değerleri

Steganografide sıklıkla kullanılan görsel (resim, metin dosyaları vb.) ve işitsel (ses, video dosyaları vb.) taşıyıcılarda oluşabilecek gürültü ve bozulmalar, gerçekleştirilmek istenilen veri gizleme ve koruma işlemlerinin fark edilmesindeki en önemli unsurlardan birisidir. Bu bağlamda yapılan daha önceki steganografi [18,19,20,21,28,29,30,35] ve stega-analiz [7,23,24,25,34] çalışmalarıda dikkate alındığında, steganografi temellerini oluşturan algılanamazlık özelliğinin de başarılı bir şekilde gerçekleşip gerçekleşmediğinide test etmek önemli bir adımı oluşturmaktadır. Bu amaçla algılanamazlık başarımının testi için SNR (Signal-to-Noise Ratio - Sinyal Gürültü Oranı) ve PSNR (Peak Signal-to-Noise Ratio - Doruk Sinyal Gürültü Oranı) değerlerinden faydalanılmaktadır [31,32,33]. Bu değerlerin nasıl hesaplandığı önceki bölüm başlıkları altında verilmiştir.

Taşıyıcı nesne (örten ortam) olarak resim, ses, video kullanılarak gerçekleştirilen steganografik yapılarda en önemli unsurlardan birisi stego (taşıyıcı + mesaj) nesne üzerinde meydana gelen olası bozulmalardır. Bu bozulma oranları taşıyıcı nesne üzerine veri saklarken değişen/değiştirilen bit sayısına göre değiklik göstermektedir.

Özellikle işitsel hassaiyeti ile öne çıkan ses ve video dosyalarında bu bozulma/değişme oranları daha da fazla önem taşımaktadır.

Bu bağlamda bahsi geçen ortamlar üzerinde yapılan steganografik çalışmalarında algılanamazlık özelliğinin verimliliği yapılan çalışmaların etkinliğini doğrudan etkilemektedir. Ses, resim, video gibi görsel ve işitsel nesler üzerinde gerçekleştirilen steganografi uygulamalarının algılanamazlık performansları SNR (Signal to Noise Ratio - Sinyal gürültü oranı) değerleri ile ölçülmektedir. SNR değerini hesaplama formülü aşağıda verilmiştir ve ölçüsü desibel (dB) dir.

Denkelemlerde kullanılmış olan;

- N: Toplam sinyal/işaret sayısını,
- İ: Sinyal dizisinin ilgili elemanını,
- X: Orijinal taşıyıcı nesneyi,
- Y: Stego nesneyi ifade etmektedir.

$$SNR_{dB} = 10 * \log_{10} \left(\frac{\sum_{i=1}^{N} (Xi)^2}{\sum_{i=1}^{N} (Xi - Yi)^2} \right)$$

Algılanamazlık testi için hesaplanan SNR değerinin 30 dB'den büyük veya eşit olması ses kalitesinin korunduğu anlamına gelmektedir.

Yine bu ortamlarda (resim, ses, video) gerçekleştirilen steganografi uygulamalarında algılanamazlık peformansının ölçümünde kullanılan diğer bir yönetmin PSNR (Peak Signal to Noise - Tepe/Doruk sinyal gürültü oranı) olduğunu daha önceki bölümde, steganalizde kullanılan ölçüm yöntemleri konu başlığında anlatılmıştı. Buna göre PSNR değerinin hesaplanmasında saklama sonucu oluşan hataların kareleri toplamının ortalaması (Mean Squared Error - MSE) değeri kullanılmaktadır. PSNR

ölçü birimi de desibel (dB) cinsindendir. MSE ve PSNR değerini hesaplanma formülü aşağıda verilmiştir.

$$MSE = \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(X_i - Y_i)^2 \qquad PSNR(dB) = 10\log_{10}\frac{X_{peak}^2}{MSE}$$

Gerçekleştirilen steganografi uygulamasında kullanılan taşıyıcı ses dosyası ve veri gizleme sonrası elde edilen stego ses dosyalarına ait elde edilen MSE, MAE, SNR ve PSNR değerleri Tablo 5.3.'de verilmiştir.

Tablo 5.3. Uygulama SNR, PSNR performans değerleri

Gizlenenen Veri		**Anahtar**		**Dosya Boyutu (Byte)**		**Performans Ölçüm Değerleri**			
Mesaj	**Boyut (Byte)**	**Key**	**Uzn.**	**Taşıyıcı**	**Stego**	**MSE**	**MAE**	**SNR**	**PSNR**
M1	648	ERD	3	1826816	1826816	0,000000	0.000016	44,74	59,67
M2	1194			1826816	1826816	0.000000	0.000025	42,80	57,74
M3	2475			1826816	1826816	0.000000	0.000042	40,59	55,53
M4	27080			1826816	1826816	0.000001	0.000136	35,45	50,39
M5	28160			1826816	1826816	0.000002	0.000221	33,35	48,29
M6	77411			1826816	1826816	0.000003	0.000374	31,05	45,99
M7	79095			1826816	1826816	0.000011	0.001456	25,15	40,09
M8	97530			1826816	1826816	0.000017	0.002175	23,41	38,35

Bu değerlerden de anşlaşılacağı üzere, taşıyıcı dosya içerisine gizlenen veri boyutu arttıkça, gürültü oranlarıda artacağından ses kalitisinde azda olsa bozulmalar olmuştur. Ancak uygulanan veri gömme algoritması sonucu kullanılan LSB bit sayılarının değişim oranları ses kalitesinde işitsel olarak algılanabilecek herhangi bir bozulmaya yol açmamıştır.

Yapılan steganografi uygulamasına ait elde edilen SNR, PSNR değerlerini hesaplayan matlab kodu şekil 5.1.'de verilmiştir.

```
function[] = Analiz_Fonksiyon(orjinal,stego)
orj  = orjinal;
stg = stego;
%MSE Hesaplama Kodu
 mse=0;
 for i=1:length(orj)
     mse=mse+(stg(i)-orj(i))^2;
 end
 mse=mse/length(orj);
 fprintf('mean squared error [MSE]: %f\n',mse);
%MAE Hesaplama Kodu
mae=0;
for i=1:length(orj)
    mae=mae+abs(stg(i)-orj(i));
end
mae=mae/length(orj);
fprintf('mean absolute error [MAE]: %f\n',mae);

%SNR ve PSNR Hesaplama Kodu
num=0;
den=0;
for i=1:length(orj)
    den=den+(stg(i)-orj(i))^2;
end
for i=1:length (orj)
    num=num+orj(i)^2;
end
SNR = 20*log10(sqrt(num)/sqrt(den));
PSNR= 20*log10(max(orj)/sqrt(mse));
fprintf('signal to noise ratio [SNR]: %f db\n',SNR);
fprintf('peak signal to noise ratio [PSNR]: %f db\n',PSNR);
end
```

Şekil 5.1. SNR, PSNR Değerleri hesaplama fonksiyonu matlab kodu

5.2.1. $\chi 2$ (ki-kare) analizi

$\chi 2$ (ki-kare) testi özellikle LSB tekniğini kullanan stego yapılar üzerinde sağlıklı sonuçlar veren ve bu alanda en yaygın olarak kullanılan steganaliz yöntemlerinden biridir. Üçüncü bölümde algoritma adımları verilen ki-kare testinin matlab uygulama kodları Şekil 5.2.'de verilmiştir.

Algoritma seçilen dosyanın % 1'den % 100'e kadar olan ses örneklerini incelemekte ve her bir yüzdesel kısım için veri gizleme olasılıklarını hesaplamaktadır. Uygulama analiz işlemini dosyanın başından sonunaa doğru yapmaktadır. Veri gizleme algoritmasının saldırgan (analist) açısından bilinmediği genel olarak varsayılmalıdır.

Uygulama kod parçasına bakıldığında ilk önce kullanıcıdan analiz etmek istediği dosya adını ve yolunu girmesi istenir. Eğer tanımladığı dosya diskte mevcut değil ise "Dosya Yok" mesajı verilir ve işlem sonlandırılır. Dosyanın olması durumunda ise ses dosyasının verisi, çerçeve hızı ve bitlik ölçüleri okunur ve analiz işlemine başlanır.

Analiz kodunda yer alan Cift(n) = (2n) sıklığı ve Tek(n)=(2n + 1) sıklığı, $0 \leq n \leq 127$ olacak şekilde Cift128 ve Tek128 iki vektör (dizi) olarak tanımlanır. İlk olarak Cift ve Tek dizilerinedeki her bir elemana 0 değeri atanır. Daha sonra algoritma ses dosyasındaki örnekleri sayar ve bu örneklerin sayısal değerinin tek/çift olmasına göre Cift ya da Tek dizisindeki karşılık gelen elemanı artırır. Teorik olarak 2n ve 2n + 1'in ses örneklerinin değerlerinin beklenen sıklığı Zn = (Cift(n) + Tek(n)) / 2'dir. Şimdi n adet kategori olduğunu yani n değer çifti olduğunu farz ediniz. 8 bitlik gri görüntüler olması durumunda, 128 kategori (256 / 2) vardır. Genelliğini kaybetmeksizin, n kategorisinde ölçülen meydana gelme sıklığı Cift(n) olacak şekilde değer çiftlerinin çift değerleri üzerinde yoğunlaşılacak. (Bir taşıyıcı nesne ve stego nesne içerisindeki 2n ve 2n + 1 piksel değerlerinin sıklığının toplamlarının aynı olduğuna yani bir taşıyıcı içerisindeki Cift(n) + Tek(n)'nın ve karşılık gelen stego görüntü içerisindeki Cift(n) + Tek(n)'e sayısına eşittir. Daha sonra Westfeld ve Pfitzmann minimum sıklık koşulu ortaya koymuştur, dolayısıyla eğer $0 \leq n \leq 127$ Aralığı için Cift(n) + Tek(n) ≤ 4 ise, Cift(n) = Tek(n) = Z(n) = 0 ve n = n - 1'dir. Diğer bir deyişle 2n ve 2n+1'in birleşik sıklığı 4'ten az ya da eşitse, 2n ve 2n + 1'in bireysel sıklık sayımları 0'a ayarlanır ve n kategorisi sayısı 1 azaltılır. Daha sonra n – 1 serbestlik derecesiyle Ki-kare istatistiği hesaplanır.

Stego nesne için Xi''nin Zi'ye yakın olması gerektiğinden dolayı X_{n-1}^2'in görece küçük olması ve Xi'nin Zi'den uzak olması gerektiğinden dolayı X_{n-1}^2'in görece

büyük olması beklenir. İşlemin son adımı, yoğunluk işlevinin üst sınır olarak X^2_{n-1} ile integralini alarak veri gizleme olasılığını, p, hesaplamaktır.

```
% Ki Kare Testi %
clear, clc, close all

calismaYeri='Stego/';
audio_name = 'stego8_lzm.wav';%'stego10K1Y70.wav';
stego_dosya_adi=strcat(calismaYeri, audio_name);
if exist(stego_dosya_adi, 'file') == false
    disp('Dosya YOK');
    return
else
   stegoinfo = audioinfo(stego_dosya_adi)
   disp('DOSYA VAR DEVAM')
end
% fs = Frame Speed = Çerçeve Hızı (frekans değeri)
[stego_nesne,fs]  = audioread(stego_dosya_adi);

% bps = BitsPerSample : Ses Örnekleri Kaç Bitten Oluşuyor [8, 16 …]
bps=stegoinfo.BitsPerSample;
%Ses dosyasından double veri tipinde okunan değerler normalize ediliyor
stego_nesne=round(2^(bps-1).*(stego_nesne))+2^(bps-1);

[satir,sutun] = size(stego_nesne); % diziyi döngü ile taramak için size alındı
yuzde = zeros(100,1);
olasilik= zeros(100,1);
kategori_say = zeros(100,1);
faktoriel= zeros(100,1);
toplamlar=zeros(100,1);
katogoriler_yarisi=zeros(100,1);

for y=1:100
   n = 2^(bps-1); % n tane kategori var
   yuzde(y) = yuzde(y) + y;
   toplam_ornek = floor((y/100)*satir*sutun);
   kacSatir = floor(toplam_ornek/sutun);

   Tek=zeros(2^(bps-1),1);   Cift=zeros(2^(bps-1),1);    kiKare=zeros(2^(bps-1),1);

   for i=1:kacSatir
      for j=1:sutun
         if rem(stego_nesne(i,j),2)==0
            Cift((stego_nesne(i,j)/2)+1)=Cift((stego_nesne(i,j)/2)+1)+1;
         else
            Tek(((stego_nesne(i,j)-1)/2)+1)=Tek(((stego_nesne(i,j)-1)/2)+1)+1;
         end
```

```
        end % for j=1:sutun
    end % for i=1:kacSatir

    Z = (Tek + Cift)/2;
    for i=1:(2^(bps-1))
        if (Tek(i)+Cift(i)) < 5
            Tek(i) = 0;
            Cift(i) = 0;
            n = n - 1;
            Z(i)=0;
        end
    end % for i=1:(2^(bps-1))

    kiKare = (Cift-Z).^2;
    for i=1:(2^(bps-1))
        if Z(i)==0
            kiKare(i) = 0;
        else
            kiKare(i) = kiKare(i)./Z(i);
        end
    end %for i=1:(2^(bps-1))

    Toplam=sum(kiKare); %C burada Ki-kare istatistiğine işaret etmektedir.
    toplamlar(y)=Toplam/2;
    katogoriler_yarisi(y)=(n-1)/2;
    try
        olasi=1-gammainc(Toplam/2,(n-1)/2); %gammainc fonksiyonun kullanarak
olasılığı hesaplarız.
        faktoriel(y)=gammainc(Toplam/2,(n-1)/2);
    catch me
    end % try
    olasilik(y) = olasilik(y) + olasi;
    kategori_say(y) = kategori_say(y) + n;
end % for y=1:100

kategoriler= zeros(2^(bps-1),1);
for i=1:(2^(bps-1))
    kategoriler(i)=2*(i-1);
end

%Elde edilen değerlerin grafik olarak gösterimi
%1: Değer Çiftleri Frekans Dağılımları
Fark=abs(Tek-Cift);
baslik='DÇ lerin Freakans Farkları;
figure1 = figure (1);
axes1 = axes('Parent',figure1);
% Uncomment the following line to preserve the Y-limits of the axes
ylim(axes1,[0 2650]);
```

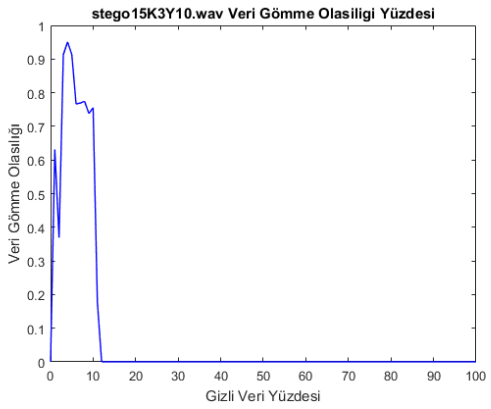
```
box(axes1,'on');
hold(axes1,'all');
figure (1),
plot(kategoriler,Fark,'LineWidth',1,'Color','blue');
xlabel('Kategoriler(i)');
ylabel('|Tek-Çift| PoVs (DÇ)');
title(baslik);

%2: Veri Gömme Olasılık Gösterimi
fig_title = [audio_name,' Veri Gömme Olasiligi Yüzdesi'];

kategori_say=cat(1,[0],kategori_say(1:100));
yuzde = cat(1, [0],yuzde(1:100));
olasilik = cat(1, [0], olasilik(1:100));

sonuc = cat(2, yuzde, olasilik, kategori_say);
toplam_kat=cat(2, toplamlar, katogoriler_yarisi);
figure (2),
plot(yuzde, olasilik,'*:');
%plot(yuzde,olasilik,'LineWidth',1,'Color','blue');
title(fig_title);  xlabel('Gizli Veri Yüzdesi'); ylabel('Veri Gömme Olasılığı');
```

Şekil 5.2. $\chi 2$ (ki-kare) testi matlab uygulama kodları

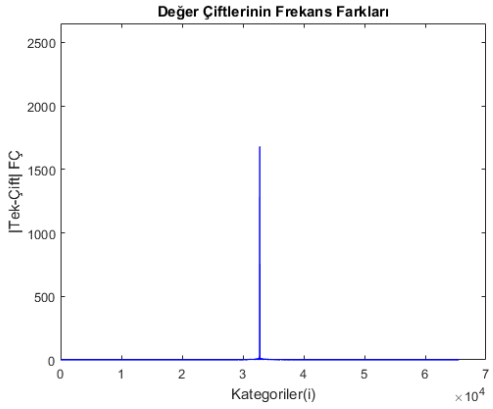
Uygulama kodunu özetleyecek olursak; ki-kare algoritmasına göre yazılan steganaliz kodu seçilen dosyayı yüzdelik dilimler halinde incelemektedir. %1‟den başlanarak %100‟e kadar tüm dosya incelenmektedir. Sıra ile önce ses dosyasının %1‟inde gizli verinin varlığı olasılıksal olarak incelenir ve 0 ile 1 arasında bir değer hesaplanır. Burada hesaplanan olasılık değeri dosyanın %1‟inde yani baş kısmında gizli verinin varlığının olasılığıdır. %1‟inde gizli veri var ise 1‟e yakın bir değer tersi ise 0‟a yakın bir değer hesaplanır. Bu şekilde %2‟sinden %100‟üne kadar araştırma devam eder. Son olarak genel tablodan dosyada ne kadar veri olduğu saptanmaya çalışılır.

Bu uygulama kodlarına göre EDstego yazılımı ile elde edilen stego nesnelerin steganalizleri yapılmış ve bu dosyalar üzerinde veri gizleme olup olmadığı test edilmiştir. Şekil 5.3.'de içerisine %10 oranda, son bitler üzerine düzgün sıralı olarak veri gizlenmiş bir ses dosyasına ait veri gömme olasılığı ve bu ses dosyasının değer çiftlerinin gösterildiği analiz sonuçları verilmiştir. Şekil 5.3.a'da görüldüğü gibi, gizlenen verinin boyutu düşük olsa bile ardı sıra gelen ses örneklerindeki değişim fazlalığı stego dosyasının algılanamazlık özelliğini yitirmesine sebep olmuştur. Ses

dosyası içerisinde hangi oranında gizli veri olma olasılığı tespit edilmiş ve bu tespit değerleri grafiğe yansımıştır. Şekil 5.4.'de uygulama içersinde kullanılan 1781 KB büyüklüğünde ve 20 sn uzunluğundaki orijinal ses dosyasına ait analizler Şekil 5.5.'de ise bu ses dosyasına daha önce dosya bilgileri (tür, uzunluk) verilmiş olan verilerin gizleme işlemi sonrası elde edilen stego nesnelere ait ki-kare analiz sonuçları gösterilmiştir. Gizli ver olarak saklanan dosya boyutlarındaki büyümeye rağmen gerçekleştirilen steganografi uygulaması sonucu elde edilen stego nesnelere yapılan ki-kare saldırısı sonucu herhangi bir veri gömme olasılığı tespit edilememiştir.

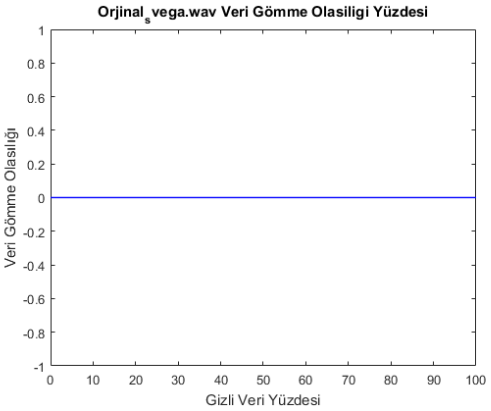


a) Stego nesne veri gömme olasılığı

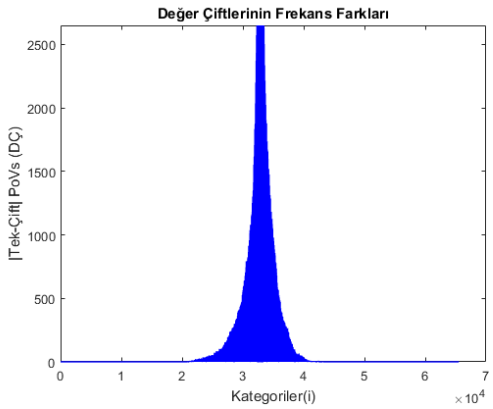


b) Stego nesneye ait değer çiftleri

Şekil 5.3. Farklı veri gömme tekniği kullanılarak elde edilen stego nesneye ait ki-kare analiz sonuçları



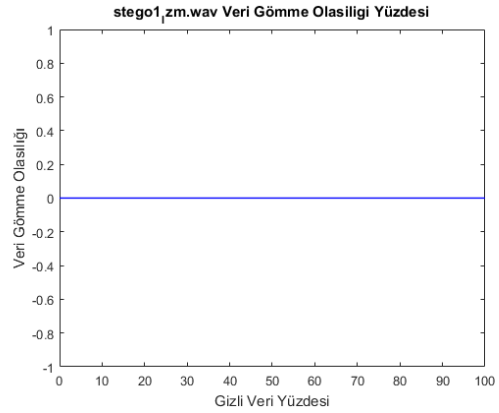
a) Orijinal ses dosyası veri gömme olasılığı



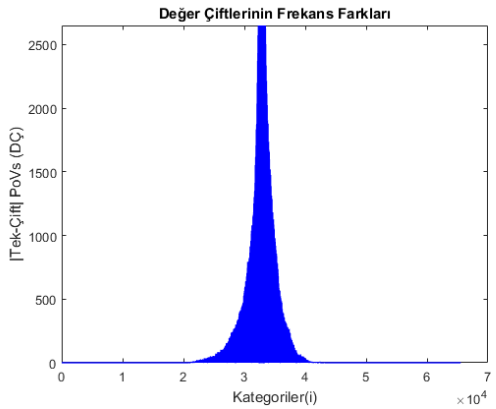
b) Orjinal ses dosyasına ait değer çiftleri

Şekil 5.4. Orjinal ses dosyasına ait ki-kare analizleri ve değer çiftlerinin gösterimi

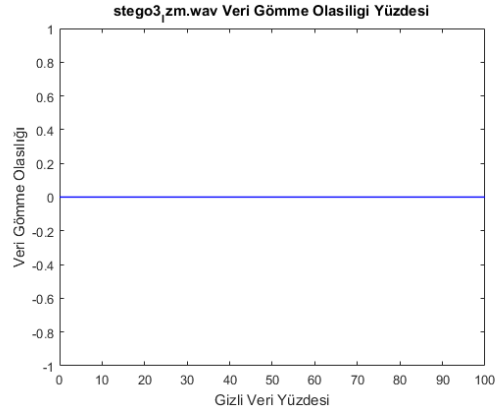
Şekil 5.4. incelendiğinde a) şeklinde verilen veri gömme olasılığı sıfır (0) olarak tespit edilmiştir. Bu dosya içerisinde henüz hiçbir veri gizlenmemiş orijinal haldedir. Yine bu dosyaya ait değer çiftleri grafiğine büyütülerek bakıldığında sağ ve sol değerlerinin olması gerektiği gibi birbirine eşit olmadığı görülmektedir.



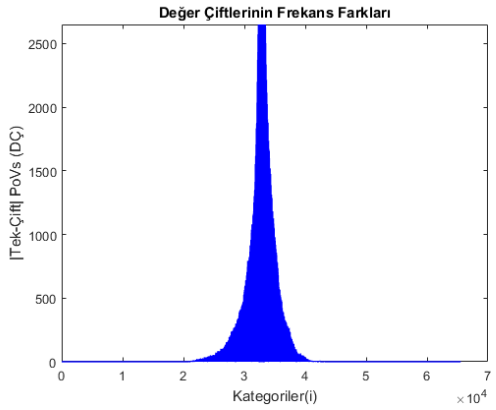
a) Stego1 nesnesi veri gömme olasılığı



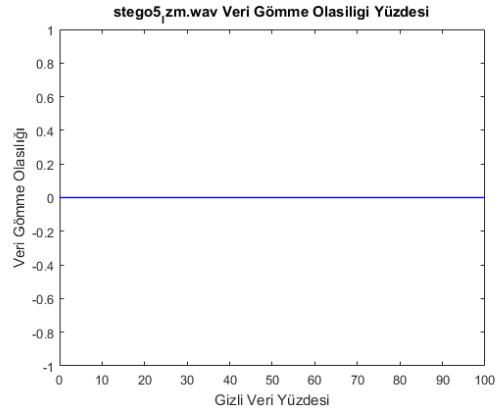
b) Stego1 dosyasına ait değer çiftleri



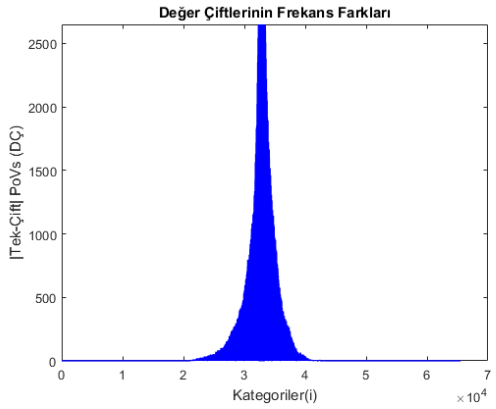
c) Stego3 nesnesi veri gömme olasılığı



ç) Stego3 dosyasına ait değer çiftleri



d) Stego5 nesnesi veri gömme olasılığı



e) Stego5 dosyasına ait değer çiftleri

Şekil 5.5. Stego nesnelere ait ki-kare analiz sonuçları

Şekil 5.5.'de verilen stego nesnelere ait ki-kare analiz sonuçlarına bakıldığında da hiçbir veri gömme olasılığının tespit edilemediği görülmektedir. Bunun sebebi ses dosyasının en az önemli bitlerine veri gizlerken uygulanan rastgele (karmaşık düzende atlayarak) veri gömme tekniğinin uygulanması ve sadece birinci seviye bitlere veri gizlenmiş olmasıdır.

5.2.2. Diğer analizler

Bu başlık altında orijinal ses dosyası ve stego ses dosyalarını karşılaştırmak ve analiz etmek için beş farklı analiz tekniği uygulanmış ve sonuçlar değerlendirilmiştir. Ki-kare testi dışında analiz teknikleri olarak:

- Histogram analizi
- Spektrogram analizi
- Genlik spektrum analizi
- Değer çiftleri karşılaştırma analizi
- Otokorelasyon analizi

Teknikleri uygulanmıştır. Analizlerin yapıldığı matlab kodu Şekil 5.6.'da verilmiştir. Şekil 5.7.'de orijinal ses dosyasına ait analiz sonuçları gösterilmiş, stego nesnelere ait analiz sonuçlarına ise Şekil 5.9. ve Şekil 5.10.'da yer verilmiştir.

Analiz kod parçasına bakıldığında görülecektirki incelenen ses dosyasının tamamı taranabileceği gibi, istenilen bir bölümü üzerinde de tarama yapılabilmektedir. Bu analiz uygulamasında tespitlerin doğru bir şekilde yapılabilmesi için orijinal ve stego ses dosyalarının tamamı analize tabi tutulmuştur. Uygulama kod parçası içerisinde açıklama satırı olarak yapılan işlemler özetlenmiştir.

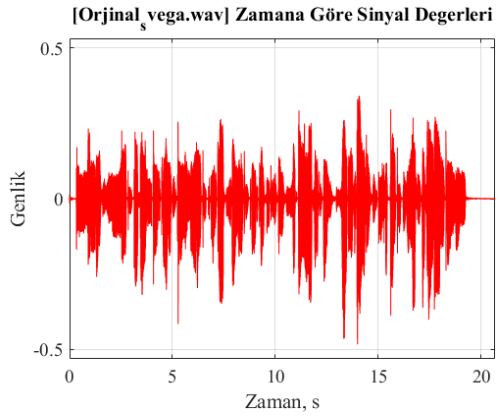
Analiz sonucu elde edilen grafik değerleri incelendiğinde, b) şekillerinde sonuç değerlerine yer verilmiş olan, sadece spektrogram analizi sonuçlarında stego nesne içerisindeki bozulma/gürültüler tespit edilebilmiştir. Bu da yapmış olduğumuz stego

uygulamasının başarısını göstermektedir. Orijinal ses dosyası ve stego ses dosyası elimizde ise bu iki yapı arasındaki farkların en kesin olarak belirleneceği yöntem Spektrogram analizi olarak karşımıza çıkmaktadır. Spektrogram analizi ses sinyalleri üzerinde meydana gelen en ufak değişikliği algılama yeteneğine sahiptir. Doğru ve hatasız bir şekilde kodlaması yapılmak suretiyle Spektrogram analizi steganalizler içerisinde uygulanacak ve doğru sonuç verecek en önemli teknik olarak değerlendirilmiştir.

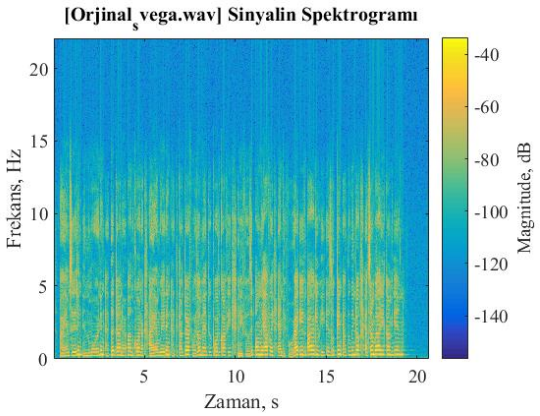
```
clear, clc, close all
% Ses dosyasını Aç ve Bir Bölümünü Al // Burada tamamı alındı
calismaYeri='Stego/';
audio_name = 'stego6_lzm.wav'
stego_dosya_adi=strcat(calismaYeri, audio_name);

[x, fs] = audioread(stego_dosya_adi, 'double');   % load an audio file
x = x(:, 1);              % ses dosyasının 1.kanal değerlerini al
N = length(x);          % signal uzunluğunu hesapla
t = (0:N-1)/fs;          % zaman vektörü (toplam süre sn)
maxt = max(t);
disp(['Sinyal Uzunluk [Örnek Sayısı] = ' num2str(N)])
disp(['Zaman Vektörü [Toplam Süre] = ' num2str(maxt)])
audio_name = (['[', audio_name, ']']);

%1 plot the signal waveform
figure(1)
plot(t, x, 'r')
xlim([0 max(t)])
ylim([-1.1*max(abs(x)) 1.1*max(abs(x))])
set(gca, 'FontName', 'Times New Roman', 'FontSize', 14)
xlabel('Zaman, s')  ylabel('Genlik') grid on
title([audio_name,' Zamana Göre Sinyal Degerleri'])

%2 Sinyal spectrogram grafiği
figure(2)
spectrogram(x, 1024, 3/4*1024, [], fs, 'yaxis')
box on
set(gca, 'FontName', 'Times New Roman', 'FontSize', 14)
xlabel('Zaman, s')  ylabel('Frekans, Hz')
title([audio_name,' Sinyalin Spektrogramı'])
%=======================================================
 h = colorbar;
set(h, 'FontName', 'Times New Roman', 'FontSize', 14)
ylabel(h, 'Magnitude, dB')
%3 Spectral analysis değerlerini hesapla
w = hanning(N, 'periodic');
[X, f] = periodogram(x, w, N, fs, 'power');
X = 20*log10(sqrt(X)*sqrt(2));
```

```
% Sinyal Spectrum Grafiği
figure(3)
semilogx(f, X, 'r')     xlim([0 max(f)])        grid on
set(gca, 'FontName', 'Times New Roman', 'FontSize', 14)
title([audio_name, ' Sinyalin Genlik Spectrumu'])
xlabel('Frekans, Hz')   ylabel('Genlik, dB')
%=======================================================
%4 Sinyal Histogram Grafiğini Çiz
figure(4)
histogram(x)
xlim([-1.1*max(abs(x)) 1.1*max(abs(x))])
grid on
set(gca, 'FontName', 'Times New Roman', 'FontSize', 14)
xlabel('Sinyal Genlik')   ylabel('Örnek Sayısı')
title([audio_name,' Sinyalin Olasılık Dağılımı'])
%=======================================================
%5 Otokorelasyon fonksiyonu Tahmini
[Rx, lags] = xcorr(x, 'coeff');
d = lags/fs;
 % Otokorelasyon Fonksiyon Grafiğini Çiz
figure(5)
plot(d, Rx, 'r')
grid on
xlim([-max(d) max(d)])
set(gca, 'FontName', 'Times New Roman', 'FontSize', 14)
xlabel('Gecikme, s')    ylabel('Otokorelasyon Katsayısı')
title([audio_name,' Sinyalin Otokorelasyon Değerleri'])
line([-max(abs(d)) max(abs(d))], [0.05 0.05],...
    'Color', 'k', 'LineWidth', 2, 'LineStyle', '--')
%=======================================================
% minimum ve maksimum değerleri hesaplanıp görüntüleniyor
maxval = max(x);    minval = min(x);
disp(['Max value = ' num2str(maxval)])
disp(['Min value = ' num2str(minval)])
% Ayrık Kosinus(DC) ve Ortalama Kare Hataları[RMS(Root Mean Square)] Değerleri
hesaplanıp görüntüleniyor
u = mean(x);   s = std(x);
disp(['Mean value = ' num2str(u)])
disp(['RMS value = ' num2str(s)])
% Dinamik Aralık Hesaplanıp Gösteriliyor
D = 20*log10(maxval/min(abs(nonzeros(x))));
disp(['Dinamik Aralık Değeri D = ' num2str(D) ' dB'])
% Tepe (Doruk) Değeri/Faktoru Hesaplanıp Gösteriliyor
Q = 20*log10(maxval/s);
disp(['Tepe/Doruk Degeri Q = ' num2str(Q) ' dB'])
% Otokorelasyon Süresi Hesaplanıp Gösteriliyor (Birbirini izleyen değerlerin arasındaki ilişki)
ind = find(Rx>0.05, 1, 'last');
RT = (ind-N)/fs;
disp(['Otokorelasyon Zamanı = ' num2str(RT) ' s'])
```

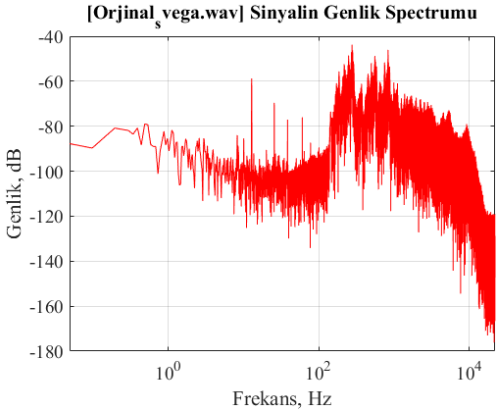
Şekil 5.6. Diğer analizlere (Histogram, Genlik, Spektrogram, DÇ, Otokorelasyon) ait analizleri matlab kod parçası



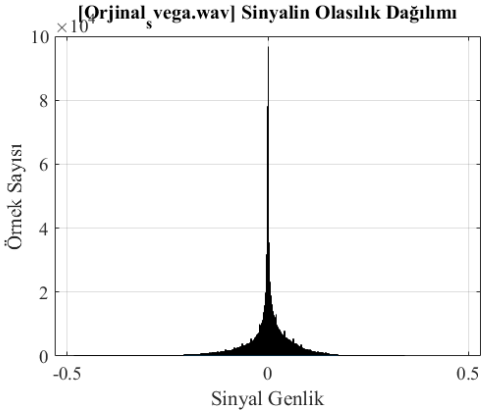
a) Orjinal nesne histogram değerleri



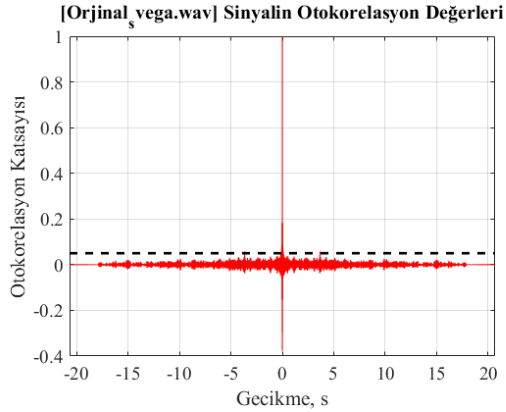
b) Orijinal nesne spektrogram değerleri



c) Orijinal nesne genlik spektrum değerleri

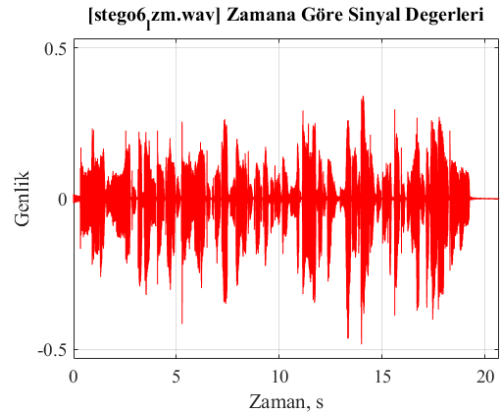


ç) Orijinal nesne değer çiftleri değerleri

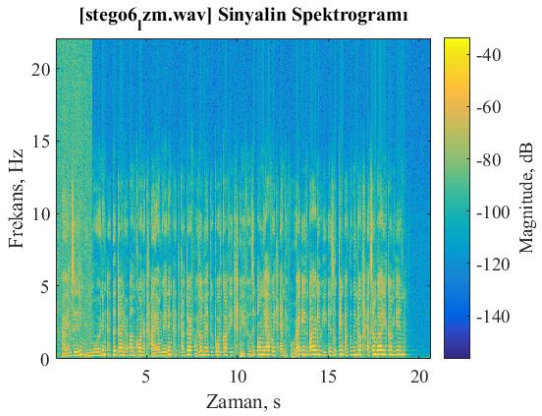


d) Orijinal nesne otokorelasyon değerleri

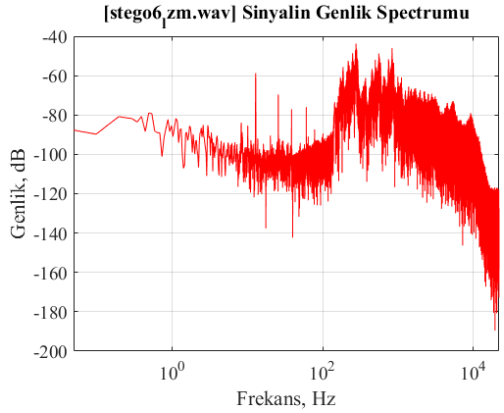
Şekil 5.7. Orjinal nesneye ait diğer (Histogram, Genlik, Spektrogram, DÇ, Otokorelasyon) analiz sonuçları



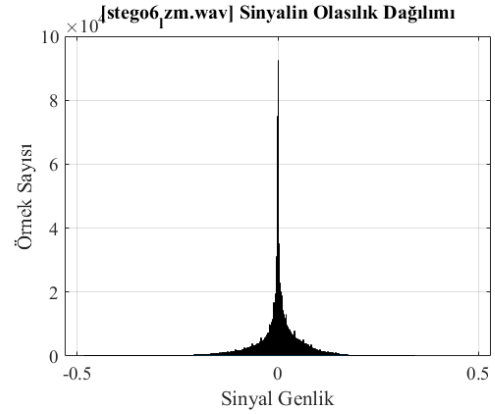
a) Stego6 nesnesi histogram değerleri



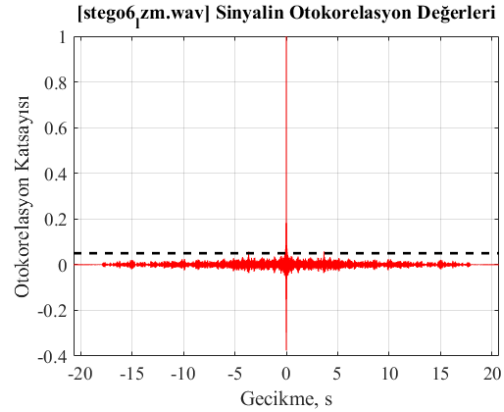
b) Stego6 nesnesi spektrogram değerleri



c) Stego6 nesnesi genlik spektrum değerleri

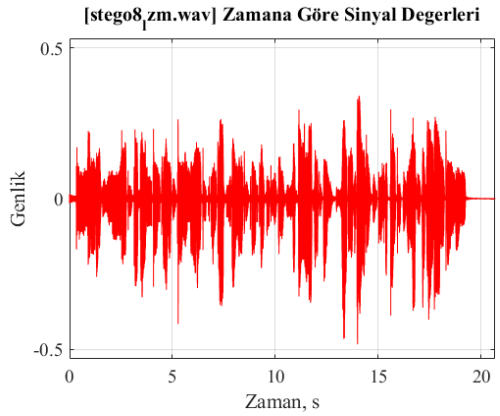


ç) Stego6 nesnesi değer çiftleri değerleri

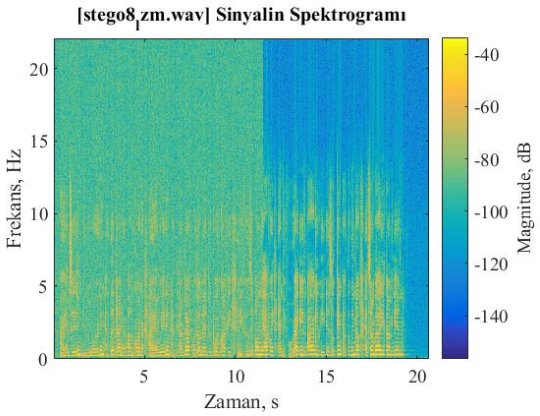


d) Stego6 nesnesi otokorelasyon değerleri

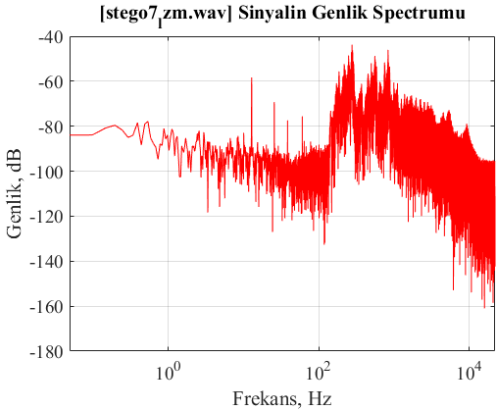
Şekil 5.8. Stego6 nesneye ait diğer (Histogram, Genlik, Spektrogram, DÇ, Otokorelasyon) analiz sonuçları



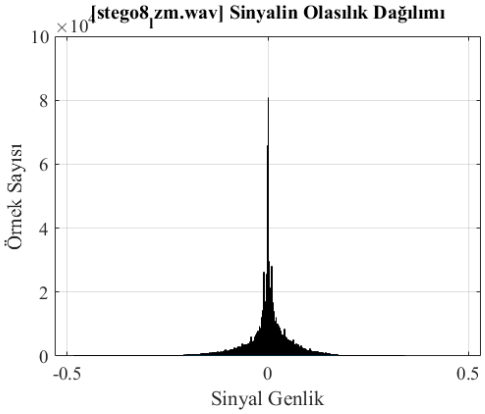
a) Stego8 nesnesi histogram değerleri



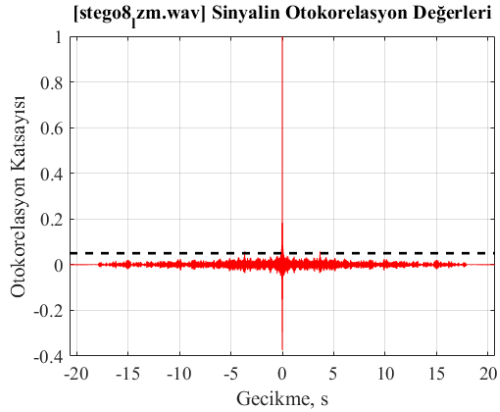
b) Stego8 nesnesi spektrogram değerleri



c) Stego8 nesnesi genlik spektrum değerleri



ç) Stego8 nesnesi değer çiftleri değerleri



d) Stego8 nesnesi otokorelasyon değerleri

Şekil 5.9. Stego8 nesneye ait diğer (Histogram, Genlik, Spektrogram, DÇ, Otokorelasyon) analiz sonuçları

5.3. Değerlendirme ve Sonuç

Tez çalışmasında taşıyıcı dosya (resim, ses) içerisine LSB tekniğini kullanarak veri saklama ve tersi adımları gerçekleştirerek stego dosya içerisinden gizli verileri çıkarma işlemi başarılı bir şekilde yapan bir steganografi uygulaması yapılmıştır.

Veri gizleme uygulaması dışında başta ki-kare analizi olmak üzere, stego nesye histogram, genlik, spektrogram, dç (değer çiftleri), otokorelasyon analizlerinin kodları matlab ortamında yazılmış ve sonuçlar değerlendirilmiştir.

Uygulamanın Türkçe dil desteği sunması ülkemizdeki bu alanda az sayıda olan steganografi ugulama eksiksikliğini gidermek adına önemli bir adım olarak değerlendirilebilir.

Geliştirilen uygulama incelenen yirmi farklı steganografi uygulaması ile karşılaştırıldığında bütün dosya formatlarına veri şifreleme ve kayıpsız veri sıkıştırma tekniklerini uygulayabilen, kodlamış bu verileri resim dosyaları (bmp, png, ico) ve ses dosyaları (wav) içerisine gizleyebilen tek steganografik uygulama olarak öne çıkmaktadır.

Geliştirilen uygulamada veri gizleme işlemi için sadece metin tabanlı dosyalar değil, bütün dosya formatları kullanılabilmektedir. Bu da EDstego uygulamasını diğer birçok steganografik uygulamadan farklı kılmaktadır.

Taşıyıcı nesne üzerinde veri saklama kapasitesini artırmak için kayıpsız veri sıkıştırma algoritmaları kullanılmış, dosya türüne ve içerdiği bilginin uzunluğuna bağlı olarak değişiklik gösteren %85'varan etkili sıkıştırma sonuçları elde edilmiştir. Uygulamanın temeli de bu sıkıştırma tekniklerinin steganografi işlemleri içerisinde kullanılarak taşıyıcı dosya üzerinde daha fazla veri saklanabilmesini hedeflemiş ve bu da başarılı bir şekilde gerçekleştirilmiştir. Mevcut Steganografik uygulamalar

içerisinde bütün dosya formatlarına veri sıkıştırması yapabilen yazılım sayısı çok azdır.

Veri güvenliğini artırmak amacıyla gizlenecek bilgilerin farklı simetrik şifreleme algoritmalarıyla şifreleme imkânı sunulmuştur. Taşıyıcı dosya içerisine gömme işleminden önce yapılan işlemler (sıkıştırma, şifreleme) gizli veri içerisine yazılıp, veri çıkarma adımında bu işlemlerin tersi uygulanarak gizli veri kayıpsız ve hatasız olarak geri elde edilmiştir.

Yapılması muhtemel steganaliz ataklarına karşı algılanamazlık ilkesini yerine getirmek için taşıyıcı dosyanın LSB bitlerinin sadece birinci seviyesine veriler gizlenmiştir. Bu gizleme işlemi taşıyıcı dosyanın ardı sıra gelen bitlerine sıralı olarak değil, karmaşık düzende atlayarak gizlenmiştir. Bu sayede yapılan ve sonuçları verilen steganaliz ataklarından da başarılı bir şekilde çıkmıştır.

Gerçekleştirilen steganografi uygulaması içinde barındırdığı özellikler sayesinde veri gizleme işleminden bağımsız olarak sadece veri sıkıştırma, sadece veri şifreleme veya her iki tekniğinde kullanıldığı kombine bir dosya elde etme imkanı sağlamaktadır. Bu özelliği ile diğer bütün yazılımlardan farklı bir yapı ortaya koymaktadır.

Gerçekleştirilen steganografi uygulaması taşıyıcı dosyaların LSB bitleri üzerinde işlem gerçekleştirmektedir. Bu tekniğin en önemli dezavantajı elde edilen stego nesneye yönelik bozma, değiştirme, yok etme saldırılarına karşı korumasız olmasıdır. Bu sebeple uygulamanın bir sonraki versiyonunda LSB kodlamaya ek olarak diğer steganografi (faz kodlama, tayf yayılması, yankı veri kodlaması) tekniklerininde kullanılması amaçlanmaktadır.

Bilgi paylaştıkça daha değerlidir ilkesi gözönüne alınarak, gerçekleştirilen uygulamaya ait veri gizleme ve analiz kaynak kodları herkese açık olacak şekilde paylaşılmıştır. Uygulamaya ait kaynak kodlarına; https://websitem.gazi.edu.tr/site/dumanertugrul/files/EDstego adresinden ulaşılabilir.

Yapılan bu tez çalışmansının ve gerçekleştrilen uygulamanın diğer araştırmacılara yardımcı olması ve ışık tutması en önemli kazanç olacaktır.

KAYNAKLAR

[1] Kahn, D., The Codebreakers, Macmillan Publishing Company, New York, 1-473, 1996.

[2] http://world.std.com/~cme/html/timeline.html., Erişim Tarihi: 15.12.2017

[3] http://www.cypher.com.au/crypto_history.htm, Erişim Tarihi: 21.12.2017

[4] Singh, S., Kod Kitabı, Klan Yayınları, İstanbul, 1-472, 2004.

[5] Stallings, W., Cryptography and Network Security, Principles and Practice, Sixth Edition, Pearson Education, New Jersey, 1-758, 2014.

[6] Dalkılıç, G., Akın, O., Anahtar Tabanlı Gelişmiş Rotor Makinesi. 7. Akademik Bilişim Konferansı., Gaziantep Üniversitesi, Gaziantep, 2005.

[7] Bar, T.H., Invitation to Cryptology, Upper Saddle River, Prentice Hall, ss: 16-19, 2002.

[8] Thomas, H. Bar, Invitation to Cryptology. Upper Saddle River, Prentice Hall, 243-314, 2002.

[9] http://en.wikipedia.org/wiki/Public-key_cryptography., Erişim Tarihi: 04.02.2018

[10] Kodaz, H., Botsalı, F.M., Simetrik ve asimetrik şifreleme algoritmalarının karşılaştırılması., Selçuk-Teknik Dergisi., 9(1):10-23, 2010.

[11] Menezes, A. J., Oorschot P. C. Van, Vanstone S. A., Handbook of Applied Cryptography. CRC Press, 2-49, 263-266, 1996

[12] Kak, A., Computer And Netwrok Security: Lecture Notes, Chapter 8, 2-18, 2018.

[13] https://www.schneier.com/academic/twofish/., Erişim Tarihi: 20.05.2017

[14] Ferguson, N., Schneir, B., Practical Cyrptography, Wiley Publishing, 59-61, 2003.

[15] https://en.wikipedia.org/wiki/Advanced_Encryption_Standard., Erişim Tarihi: 22.05.2017

[16] Demir, N., Dalkılıç, G., Anahtar bağımlı bir şifreleme algoritması (IRON). Akademik Bilişim Konferansı., Kastamonu, 2007.

[17] http://bilgisayarkavramlari.sadievrenseker.com., Erişim Tarihi: 21.05.2017

[18] Şahin, F., Modern blok şifre algoritmalarının incelenmesi., İstanbul Teknik Dergisi., 7(26): 23-40, 2015.

[19] Sakalli M.T., Moder blok şifre algoritmalarının gücü., researchgate Online Dergi

[20] Keliher, L., Linear Cryptanalysis of Substitution-Permutation Networks, Ph.D. Thesis, 2002.

[21] Dandekar, A. K., Pradhan, S., Ghormade, S., Design of AES-512 algorithm for communication network., International Research Journal of Engineering and Technology (IRJET), 3(5): 438-443, 2016.

[22] Rajalakshmi, M.R.K., Abarna, N., A modıfıed approach to ımprovıse the effıcıency of des and aes usıng 1024-bıt key., International Research Journal of Engineering and Technology (IRJET), 3(5): 2754-2760, 2016.

[23] Thakur, J., Kumar, N., DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis., International Journal of Emerging Technology and Advanced Engineering, 1(2): 6-12, 2011

[24] Obaid, Z., Sabonchi, A., Akay, B., Klasik kriptoloji yöntemlerinin karşılaştırılması., Engineering Sciences, 11(4):100-108, 2016.

[25] Lin, C. H., Lee T. C., A confused document encryption scheme and its ımplementation., Elsevier Computer & Security, 17(6):543-551, 1998

[26] Gambhir, A., Khara, S., Integrating RSA Cryptography & Audio Steganography. International Conference on Computing, Communication and Automation., 481-483, 2016.

[27] Günden Ü., Şifreleme algoritmalarının performans analizi. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar ve Bilişim Mühendisliği Bölümü, Yüksek Lisans Tezi, 2010.

[28] Atıcı, M.A., Steganografik yaklaşımların incelenmesi, tasarımı ve geliştirilmesi. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, 2007.

[29] Shirali-Shahreza, M., Text steganography by changing words spelling, Proceedings of the 10th International Conference on Advanced Communication Technology (ICACT 2008), Phoenix Park, Korea, 1912-1913., 2008.

[30] Satir, E., Bilgi güvenliği için metin steganografisinde yeni bir yaklaşım. Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Doktora Tezi, 2013.

[31] Salomon, D., Coding for Data and Computer Communications, Springer Science + Business Media Inc., USA, 2005.

[32] Johnson, N. F., Duric Z. ve Jajodia **S.,** Information Hiding :Steganography and Watermarking - Attacks and Countermeasures, Boston, 2001.

[33] Yeh, W. H., Hwang, J.J., Hiding digital information using a novel system scheme., Computers & Security, 20(6): 533-538, 2001.

[34] Gutub, A., Fattani, M., A novel Arabic text steganography method using letter points and extensions, WASET International Conference on Computer, Information and Systems Science and Engineering (ICCISSE), Vienna, Austria, 28–31, 2007.

[35] Zaker, N., Hamzeh, A., A novel steganalysis for TPVD steganographic method based on differences of pixel difference histogram, Multimed Tool Appl., 2011.

[36] http://mesutpiskin.com/blog/renk-uzaylari.html., Erişim Tarihi: 15.02.2018

[37] http://colorizer.org., Erişim Tarihi: 15.02.2018

[38] Asmara, R.A., Agustina, R., Hidayatulloh., Comparison of Discrete Cosine Transforms (DCT), Discrete Fourier Transforms (DFT), and Discrete Wavelet Transforms (DWT) in Digital Image Watermarking, International Journal of Advanced Computer Science and Applications (IJACSA), 8(2): 245-249, 2017.

[39] Gorodetski, V.I., Popyack, L.J., Samoilov, V., Skormin V.A., SVD-Based Approach to Transparent Embedding Data into Digital Images, Information Assurance in Computer Networks. 263-274, 2001.

[40] Fkirin, A., Attiya, G., Ayman, El-Sayed., Steganography literature survey, Classification and comparative study. Communications on Applied Electronics (CAE), Foundation of Computer Science FCS, New York, USA, 5(10): 13-22, 2016.

[41] Aydın. S., Memiş, M., Elbaşı, e., Digital Image Watermarking Method in Multi Level DWT. 16th Signal Processing, Communication and Applications Conference (IEEE), Aydın, 1-4, 2008.

[42] Bangera, K.N., Paddambail, Y., Reddy, S., Shivaprasad, G., Multilayer security using rsa cryptography and dual audio steganography. 2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT), India, 492-495, 2017.

[43] Sagiroglu, S., Tunckanat, M., A secure ınternet communication tool. Turkish Journal of Telecommunications, 1(1):40-46, 2002.

[44] Salomon, D., Giovanni,M., Handbook of Data Compression 5th Edition, Springer., New York, USA, 2010.

[45] Bender, W., Gruhl, D., Morimoto, N., and Lu, A. Techniques for data hiding. IBM Syst. J., 35(3&4):313-336, 1996.

[46] Durdu, A., Sırörtülü ses dosyalarının yapay zeka yöntemleri yardımıyla çözümlenmesi. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Elek. Ve Bilg. Eğitimi Anabilim Dalı, Yüksek Lisans Tezi, 2010.

[47] Cvejic, N., Seppanen, T., Increasing the capacity of LSB-based audio steganography. Multimedia Signal Processing, IEEE Workshop, St. Thomas, Virgin Islands, USA, 336- 338, 2002.

[48] Gopalan, K., Audio steganography using bit modification. International Conference on Multimedia and Expo, Baltimore, Maryland, 629-632, 2003.

[49] Djebbar, F., Ayad, B., Meraim, K.A., Hamam, H., Comparative study of digital audio steganography techniques. EURASIP Journal on Audio, Speech, and Music Processing. 6(1), 1-16, 2012

[50] Singh, P., A Comparative Study of Audio Steganography Techniques. International Research Journal of Engineering and Technology (IRJET). 3(4), 580-585, 2016.

[51] http://andacmesut.trakya.edu.tr/bgt/., Erişim Tarihi: 03.12.2017

[52] Katzenbeisser, S., Petitcolas, F.A.P., Information Hiding Techniques for Steganography, Artech House, INC. 1-273, 2000.

[53] Fridrich, J., Du R., Meng, L., Steganalysis of LSB Encoding in Color Images., Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conferece, New York City, USA, 3: 12791282, 2000.

[54] Fridrich, J., Minimizing the embedding impact in steganography, Proceeding of the 8th Workshop on Multimedia and Security, Geneva-Switzerland, 2-10, 2006.

[55] Aktaş, F., Steganaliz yöntemleri kullanılarak resim içerisindeki saklı bilgilerin tespit edilmesi. Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Ve Elektronik Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, 2011.

[56] https://veribilimcisi.com/2017/07/14/mse-rmse-mae-mape-metrikleri-nedir/., Erişim Tarihi: 04.11.2017

[57] Takaoğlu, F., DWT ve DCT steganografide performans analizi. İstanbul Aydın Üniversitesi, Fen bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, 2016.

[58] Devi, R.R., Pugazhenthi, D., Ideal sampling rate to reduce distortion in audio steganography. Procedia Computer Science, 85, 418-424, 2016.

[59] Durdu, A.,Özcerit, A.T., Güvenli iletişim için yeni bir veri gizleme algoritması. Uluslararası Bilgi Güvenliği Mühendisliği Dergisi, 1(1), 32-36, 2015.

[60] Fridrich, J., Goljan, M., Practical steganalysis of digilal images state of the art. Proc. SPIE Photonics West, 4675, 1-13, 2002.

[61] Atıcı, M.A., Sağıroğlu S., Development of a new folder lock appraoch and software based on steganography. Journal of the Faculty of Engineering and Architecture of Gazi University. 31(1), 129-144, 2016.

[62] Fraczek, W., Mazurczyk, W., Szczypiorski, K., Hiding information in a stream control transmission protocol. Computer Communications (ELSEVIER)., 35(2), 159-169, 2012.

[63] Kaur, S., Bansal, S., Bansal, R. K., Steganography and classification of image steganography techniques. 2014 International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, New Delhi, India, 870-875, 2014.

[64] https://en.wikipedia.org/wiki/Steganography_tools., Erişim Tarihi: 18.12.2017

[65] https://tr.wikipedia.org/wiki/ASCII., Erişim Tarihi: 08.09.2017

[66] Carus, A., Mesut, A., Fast text compression using multiplies dictionaries. Information technology journal. 9(5), 1013-1021, 2010.

[67] Satir, E., Isik, H., A compression-based text steganography method. The Journal of Systems and Software, Elsevier Sience., 85(10), 2385-2394, 2012.

[68] Wang, Z., Yang, H., Cheng, T., Chang, C., A high-performance reversible data-hiding scheme for LZW codes. The Journal of Systems and Software, Elsevier Sience., 86(11), 2771-2778, 2013.

[69] Al-Bahadili, H., A novel lossless data compression scheme based on the error correcting Hamming codes, Computers & Mathematics with Applications, 56(1), 143-150, 2008.

[70] http://marknelson.us/1989/10/01/lzw-data-compression/., Erişim Tarihi: 18.11.2017

[71] https://www.pc-audiophile.com/sayisal-ses-digital-audio-pcm-ve-dsd-hakkinda-ozet/., Erişim Tarihi: 01.12.2017

[72] http://soundfile.sapp.org/doc/WaveFormat/., Erişim Tarihi: 03.12.2017

[73] Desoky, A., 2009, Listega: list-based steganography methodology, International Journal of Information Security, 8(4), 247-261.

[74] Hassan, M.D., Steganaliz yaklaşımlarının karşılaştırılması. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, 2008.

[75] Westfeld, A., Pfitzmann, A., "Attacks on Steganographic Systems", Proceedings of the Third International Workshop Information Hiding, Dresden, Germany, 61-76, 2000.

[76] Provos, N., "Defending Against Statistical Steganalysis", 10th USENIX Securit Symposium, Washington, 323-335, 2001.

ÖZGEÇMİŞ

Ertuğrul Duman, 1983 yılında Bayburt'da doğdu. İlk, orta ve lise eğitimini Antalya'da tamamladı. 1999 yılında Gazi Lisesi'nden mezun oldu. 2004 yılında Sakarya Üniversitesi Bilgisyar Mühendisliği Bölümünü bitirdi. 2005 - 2007 yılları arasında Kocaeli Üniversitesinde yazılım uzmanı olarak çalıştı. 2004 - 2009 yılları arasında sakarya ve ankarada özel yazılım firmalarında yazılım uzmanı olarak görev yaptı. Kurumsal kaynak planlama, insan kaynakları yönetimi ve personel bilgi sistemleri alanında uygulama yazılımları geliştirdi. Yine bu yıllarda YÖK (Yükseköğretim Kurulu) merkezi personel bilgi sistemi programının yazılmasında görev aldı. Bu sistem daha sonra YÖK ve TUİK (Türkiye İstatistik Kurumu) tarafından geliştirilerek bugünkü YÖKSİS halini almıştır. 2010 yılında askerlik görevini tamamladı ve akabinde Gazi Üniversitesi'nde öğretim görevlisi olarak çalışmaya başladı. Hala bu kurumdaki görevine devam etmektedir. Sakarya Üniversitesi Bilgisayar ve Bilişim Mühendisliği Bölümü'nde yüksek lisans eğitimini bu tez çalışması ile tamamlamıştır. Evli ve iki çokcuk babasıdır.