T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÖRÜNTÜ STEGANOGRAFİSİNDE YENİ BİR YÖNTEM

Özcan ÇATALTAŞ

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Ocak-2018
KONYA

TEZ KABUL VE ONAYI

Özcan ÇATALTAŞ tarafından hazırlanan "Görüntü Steganografisinde Yeni Bir Yöntem" adlı tez çalışması 25/01/2018 tarihinde aşağıdaki jüri tarafından oy birliği / ~~oy çokluğu~~ ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Yrd. Doç. Dr. Abdullah Erdal TÜMER	.....
Danışman Yrd. Doç. Dr. Kemal TÜTÜNCÜ	.....
Üye Yrd. Doç. Dr. Murat CEYLAN	.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mustafa YILMAZ
FBE Müdürü

Bu tez çalışması Öğretim Üyesi Yetiştirme Programı tarafından 2016-ÖYP-051 nolu proje ile desteklenmiştir.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Özcan ÇATALTAŞ

Tarih: 25.01.2018

ÖZET

YÜKSEK LİSANS TEZİ

GÖRÜNTÜ STEGANOGRAFİSİNDE YENİ BİR YÖNTEM

Özcan ÇATALTAŞ

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Kemal TÜTÜNCÜ

2018, 84 Sayfa

Jüri
Yrd. Doç. Dr. Abdullah Erdal TÜMER
Yrd. Doç. Dr. Kemal TÜTÜNCÜ
Yrd. Doç. Dr. Murat CEYLAN

Son yıllarda teknolojinin hızlı ilerleyişi ve yaygınlaşmasıyla teknolojinin getirdiği yenilikleri kullanmamak imkânsız hale gelmiştir. Bu yeniliklerden olan telefon, bilgisayar ve internet kullanımı yaygınlaştıkça insanların haberleşme ve bilgi güvenliği konusunda şüpheleri de artmıştır. Bu nedenle bilgi güvenliği çağımızın en önemli konularından biri haline gelmiştir.

Klasik güvenlik yaklaşımlarında kullanılan şifreleme yani kriptografi algoritmaları bilginin gizliliğini sağlamaktadır. Fakat son zamanlarda giderek daha fazla üzerinde durulan "fark edilemezlik" ilkesini sağlayamamaktadır. Çünkü şifrelenmiş bir metinin üzerine yoğunlaşıldığı zaman gelişmiş bilgisayarlar sayesinde çözülebilmektedir. Dolayısıyla burada anahtar nokta "şüphe uyandırma" durumudur. Bu sebeple, gizli bir mesajın varlığının sezilememesini öncelikli hedefe koyan steganografi ve damgalama yöntemleri özellikle 2000'li yıllardan sonra ilgi odağı konumundadır.

Bu tez çalışmasında uzamsal düzlemde en düşük değerlikli bit değiştirme yöntemine dayalı yeni bir görüntü steganografi yöntemi önerilmiştir. Önerilen yöntemi, steganografi'nin temel kriterlerinden biri olan güvenlik açısından geliştirmek için Rivest-Shamir-Adleman(RSA) şifreleme algoritması, gizlenebilecek mesaj kapasitesini artırmak için ise Lempel-Ziv-Welch, Arithmetic, Move-to-Front ve Deflate kayıpsız sıkıştırma algoritmaları önerilen yöntem ile birleştirilmiştir. Önerilen yöntemi ve mevcut en düşük değerlikli bit değiştirme yöntemini uygulamak için MATLAB GUI ortamında gizleme ve çıkarma için iki arayüz tasarlanmıştır. Sonuçlar incelendiğinde, önerilen yöntem ile steganografinin üç kriterinde de iyileşme gösterildiği görülmüştür.

Anahtar Kelimeler: Görüntü Steganografi, Kriptografi, LSB Değiştirme Yöntemi, Veri Gizleme Yöntemleri, Veri Sıkıştırma Yöntemleri

ABSTRACT

MS THESIS

A NEW METHOD IN IMAGE STEGANOGRAPHY

Özcan ÇATALTAŞ

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN ELECTRICAL-ELECTRONIC ENGINEERING

Advisor: Asst. Prof. Dr. Kemal TÜTÜNCÜ

2018, 84 Pages

Jury

Asst. Prof. Dr. Abdullah Erdal TÜMER
Asst. Prof. Dr. Kemal TÜTÜNCÜ
Asst. Prof. Dr. Murat CEYLAN

In recent years, with the rapid progress and spread of technology, it has become impossible not to use the innovations brought by the technology. As the use of telephone, computer and internet became more and more common, the doubts about people's communication and information security have increased. For this reason, information security has become one of the most important issues of our time.

Cryptography algorithms used in classical security approaches provide the confidentiality of information. But in recent times it has been unable to achieve the principle of “unnoticeable” which is increasingly emphasized. Because when an encrypted text is focused on, it can be solved by advanced computers. So the key point here is “suspicion”. For this reason, steganography and watermarking methods, which put the invisibility of the existence of a confidential message as a primary goal, have become a focus of interest especially after 2000's.

In this thesis, a new image steganography method based on the least significant bit substitution method in the spatial plane is proposed. The proposed method is combined with the Rivest-Shamir-Adleman(RSA) encryption algorithm to improve the security of the steganography, which is one of the requirements of steganography, and the Lempel-Ziv-Welch, Arithmetic, Move-to-Front and Deflate lossless compression algorithms to increase the message capacity that can be hidden. To implement the proposed method and the existing least significant bit substitution method, two interfaces for embedding and extracting processes has been designed with MATLAB GUI environment. When the results were examined, it was seen that the proposed method showed improvement in all three criteria of steganography.

Keywords: Cryptography, Data Compression Methods, Data Hiding Methods, Image Steganography, LSB Substitution Method

ÖNSÖZ

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

Ayrıca tez çalışmam süresince bana her zaman moral veren ve destek olan değerli eşime sonsuz teşekkür ederim.

Özcan ÇATALTAŞ
KONYA-2018

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

Simgeler

φ(n): Totient fonksiyonu
e: Genel anahtar
d: Özel anahtar
$\mu_{x,y}$: Piksel yoğunluk ortalaması,
$\sigma_{x,y}$: Standart sapma
$C_{1,2}$: Katsayı

Kısaltmalar

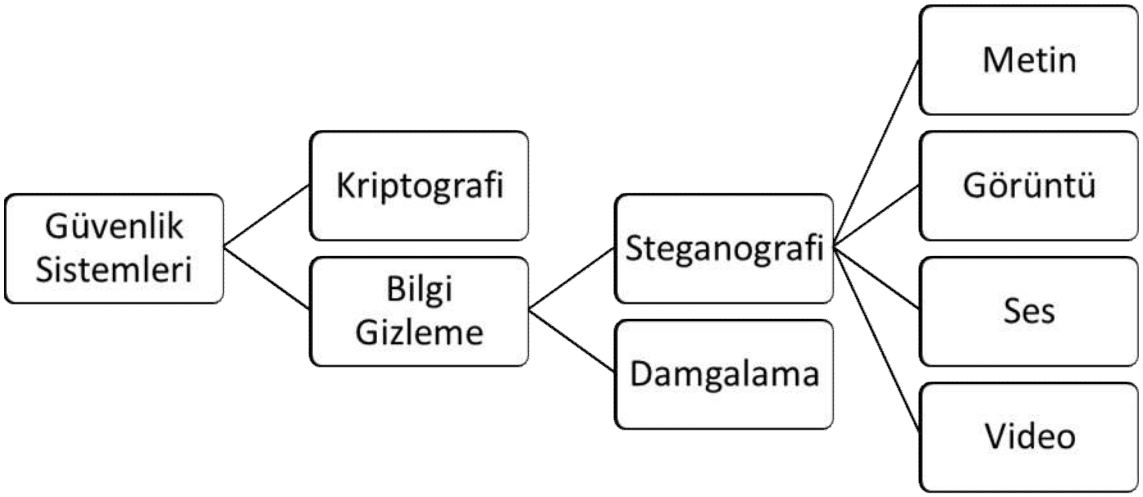
RDH: Reversible Data Hiding (Geri Çevrilebilir Veri Gizleme)
LSB: Least Significant Bit (En Düşük Değerlikli Bit)
MSB: Most Significant Bit (En Yüksek Değerlikli Bit)
DES: Data Encryption Standard (Veri Şifreleme Standardı)
AES: Advanced Encryption Standard (İleri Sıkıştırma Standardı)
RSA: Rivest-Shamir-Adleman
LZW: Lempel-Ziv-Welch
MTF: Move to Front (Öne Taşı)
MSE: Mean Square Error (Ortalama Karesel Hata)
PSNR: Peak Signal to Noise Ratio (Tepe Sinyal Gürültü Oranı)
AD: Average Difference (Ortalama Fark)
NAE: Normalized Absolute Error (Normalize Mutlak Hata)
NCC: Normalized Cross Correlation (Normalize Çapraz Korelasyon)
SSIM: Structural Similarity Index Measure (Yapısal Benzerlik İndeksi)
NIST: National Institute of Standard and Technology (Ulusal Standartlar ve Teknoloji Enstitüsü)
DCT: Discrete Cosine Transform (Ayrık Kosinüs Dönüşümü)
DWT: Discrete Wavelet Transform (Ayrık Dalgacık Dönüşümü)

1. GİRİŞ

1.1. Veri Gizleme Teknikleri

Günümüzde güvenlik, herhangi bir iletişim sisteminde yer alan en kritik faktör olarak düşünülmektedir. Bu tür sistemlerde güvenlik ilgili sorunlar bütünlük, gizlilik, kimlik doğrulamadır. Bu tür sorunların dikkatle ele alınması gereklidir. İletişim sisteminde olabilecek saldırılardan orijinal bilgileri korumak için veri gizleme teknikleri uygulanmaktadır. Bu teknikler, Dönüştürülebilir Veri Gizleme yani RDH (Reversible Data Hiding) çatısı altında gruplanmaktadır. Dönüştürülebilir veri gizleme tekniklerinin en çok bilinen iki tanesi ise veri şifreleme (kriptografi) ve veri gizleme (steganografi)'dir. Şifreleme ve veri gizleme, veri korumanın iki tekniğidir. Veri gizleme teknikleri, açıklanmasını istemediğimiz orijinal verileri kabul edilebilir değişiklikler getirerek tüm medyayı kapsar şekilde yerleştirirken, şifreleme teknikleri düz metin verilerini okunamayan biçime, yani şifreli metin haline dönüştürür. Bununla birlikte, gizli mesajları iletmek için veriyi bir dijital ortama karıştırmak faydalı olacaktır. (Patil ve Katariya, 2015)

Resim şifrelemek için RDH'yi uygulayabiliyorsak, bunun üzerinden bazı iyi uygulamalar üretilebilir. Örneğin: bir veri merkezinde depolanan tıbbi resim veri tabanını ve daha sonra bazı gösterimlerin tıbbi bir görüntünün şifrelenmiş versiyonuna gömülü olabileceğini varsayalım. Sunucu, orijinal içeriğin bilgisi olmaksızın gösterimleri kullanarak görüntüyü yönetebilir veya bütünlüğünü doğrulayabilir. Bu, hastanın mahremiyetini koruyacaktır. Aynı zamanda, bir doktor, şifreleme anahtarı kullanarak daha ileri teşhis için görüntünün şifresini çözebilir ve geri yükleyebilir. Resimlerde veya videolarda saklanabilen geri döndürülebilir veriler, gömülü verilerin çıkartılmasından sonra kayıp vermeden orijinal içeriğin geri kazanılacağı bir tekniktir. Bu teknik, orijinal kapağın bozulmamasının gerektiği tıbbi, askeri ve hukuk adli tıp gibi çeşitli alanlarda yaygın şekilde kullanılabilir. Yakın zamanda, RDH için genel bir çerçeve oluşturan birçok yeni RDH tekniği geliştirilmiştir. İlk önce orijinal taşıyıcı medyasının özelliklerini çıkararak ve daha sonra bunları kaybetmeden sıkıştırarak çalışır; sıkıştırma sonucu açılan alana ek veri gömülerek ekstra alan kazanılabilir. Önceki tüm RDH yöntemleri, şifreli görüntülerden geri dönüşümlü olarak yer açarak verileri gömmektedir. Bu da verilerin ayıklanması ve/veya resim geri yüklenirken bazı hatalara neden olabilmektedir. Bu sistemin amacı, internet üzerinden oldukça hassas bilgilerin güvenli bir şekilde iletilmesini sağlamaktadır.



Şekil 1.1. Veri gizleme yöntemleri

1.1.1. Kriptografi

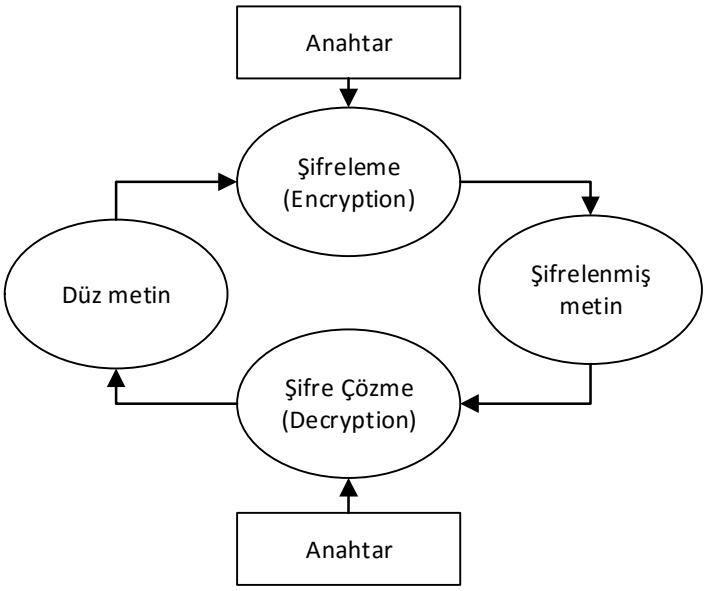
Kriptografi, şifreli metin (cipher text) adı verilen kodlanmış metin üretmek için kullanılan gizli yazma sanatıdır. Burada, kodlanmış metin üretme işlemine şifreleme (Encryption), kodlanmış metinden tekrar orijinal metni elde etme işlemine ise şifre çözme (Decryption) adı verilir (Şekil 1.2). Steganografi ve Kriptografi birbiriyle yakından ilişkilidir. Kriptografi, mesajları anlaşılmaz şekilde karıştırır. Kriptografi için Kerckhoffs yasasına göre, bir kriptografi sisteminin kalitesi gizli anahtar adı verilen küçük bir mesaj parçasına bağlıdır. Ayrıca bu bilgi parçası gizli bir mesajın varlığını ele vermemelidir. Bir mesajı çıkarmak, sadece onu ortaya çıkarmak için gereken anahtar bilgisi ile mümkün olmalıdır. Kriptografi, gizlilik anlamında Yunanca kriptolardan türetilen bilgi güvenliği bilimidir. Mikro nokta, görüntüleri kelimeleri birleştiren ve depolamada veya geçişte bilgiyi gizlemenin diğer yolları gibi teknikleri içerir. Kriptografinin çeşitli hedefleri şöyledir:

a) Gizlilik (Confidentiality): Bilgi, kimse tarafından anlaşılamamalıdır.

b) Bütünlük (Integrity): Bilgi, gönderen ile amaçlanan alıcı arasındaki depolamada veya geçişte değiştirilememelidir.

c) Reddedilmezlik (Non-repudiation): Bilginin üreticisi/göndereni daha sonraki bir aşamada bilginin oluşturulmasında veya iletilmesindeki niyetlerini inkâr edemez.

d) Doğrulama: Gönderen ve alıcı, birbirlerinin kimliğini ve bilgilerin orijinini / varış yerini teyit edebilir.



Şekil 1.2. Şifreleme akış diyagramı

Kriptografi algoritmaları, bilgi güvenliğinde önemli rol oynamaktadır. Simetrik ve asimetrik anahtar şifrelemesi olarak ikiye ayrılabilirler. Simetrik anahtar şifrelemesinde, verileri şifrelemek ve şifreyi çözmek için yalnızca tek bir anahtar kullanılır. Anahtar, iki taraf arasında iletimden önce dağıtılmalıdır. Anahtar, şifrelemede ve şifre çözmede önemli bir rol oynamaktadır. Zayıf bir anahtar algoritmada kullanılıyorsa, veri kolayca çözülebilir. Anahtarın boyutu, simetrik anahtar şifrelemesinin gücünü belirler.

Simetrik algoritmalar iki tiptir: blok şifreleme ve akış şifreleme. Blok şifreleri, gruplar halinde veya bloklar halinde çalışır. Veri Şifreleme Standardı (DES), Gelişmiş Şifreleme Standardı (AES) ve Blowfish örnek olarak verilebilir. Akış şifreleri aynı anda tek bir bitle çalışır. RC4, bir akış şifreleme algoritmasıdır. Şifreleme algoritmaları, pil gücü, CPU süresi, vb. gibi önemli bilgi işlem kaynaklarını tüketir.

Asimetrik anahtar (veya açık anahtar) şifreleme, anahtar dağıtımı sorununu çözmek için kullanılır. Asimetrik anahtar şifrelemesinde iki anahtar kullanılır; Özel anahtarlar ve ortak anahtarlar. Ortak anahtar, şifreleme için kullanılır. Şifre çözme için ise özel anahtar kullanılır (ör. Dijital İmzalar). Genel anahtar herkes tarafından bilinir, ama özel anahtar yalnızca kullanıcı tarafından bilinir. İletimden önce bunları dağıtmaya gerek yoktur. Asimetrik şifreleme teknikleri, daha fazla hesaplama işlem gücü gerektirdiği için simetrik tekniklerden yaklaşık 1000 kat daha yavaştır.

1.1.1.1. Kriptografi teknikleri

Çeşitli şifreleme teknikleri aşağıdaki gibidir:

a) DES (Veri Şifreleme Standardı): DES tanımlandığı gibi bir blok şifreleme algoritmasıdır. NIST (Ulusal Standartlar ve Teknoloji Enstitüsü) tarafından yayınlanan ilk şifreleme standardıdır. Simetrik bir algoritma, şifreleme ve şifre çözme için aynı anahtarın kullanıldığı anlamına gelmektedir. Bu yüzden DES'de 64-bit'lik bir anahtar kullanır. 64 bitten 56 biti, tam şifreleme dönüşümünü belirleyen bağımsız anahtarı oluşturken; kalan 8 bit ise hata algılama için kullanılır. Hem anahtar genişletme kısmı hem de şifre bölümünde altı farklı permütasyon işlemi yapılmıştır. DES algoritmasının şifresini çözme, şifrelemeye benzer işlemlerdir. Çıktı, 64-bitlik bir şifre metni bloğudur. Birçok saldırı ve yöntem DES'in güvenlik zayıflıklarını ortaya çıkarmıştır.

b) 3DES (Üçlü DES): 3DES, Veri Şifreleme Standardının bir geliştirmesidir. 64 bitlik bloklardan oluşan 192 bitlik anahtar kullanır. Şifreleme yöntemi, özgün DES'de bulunana benzer ancak şifreleme seviyesini ve ortalama güvenli zamanı artırmak için 3 kez DES uygulanmıştır. 3DES diğer blok şifreleme yöntemlerinden daha yavaştır.

c) AES (Gelişmiş Şifreleme Standardı): AES, Rijndael'in algoritması olarak da bilinir, simetrik bir blok şifreleme yöntemidir. Bilgisayar işlem gücündeki ilerleme nedeniyle DES'nin güvenli olmadığı anlaşılmıştır. Bu nedenle NIST, ABD devlet kurumları tarafından askeri olmayan bilgi güvenliği uygulamalarında kullanılabilecek DES yerine yeni bir standart bulmayı amaçlamıştır. Böylece AES yöntemi ortaya çıkmıştır. AES, 128, 192 veya 256 bitlik simetrik anahtarları kullanarak 128 bitlik veri bloklarını şifreleyebilir. 128, 192 veya 256 bitlik değişken anahtar uzunluğuna sahiptir fakat varsayılan değer 256'dır. Anahtar boyutuna bağlı olarak 10, 12 ve 14 turda 128 bitlik veri bloklarını şifreler. AES ile şifreleme hızlı ve esnektir. Çeşitli platformlarda özellikle küçük cihazlarda uygulanabilir. AES birçok güvenlik uygulaması için test edilmiştir.(Juneja ve Sandhu, 2013)

d) Blowfish: Bu algoritma, en çok kullanılan genel şifreleme algoritmalarından biridir. Blowfish 1993 yılında Bruce Schneider tarafından mevcut şifreleme algoritmalarına hızlı bir alternatif olarak tasarlanmıştır. Blowfish, 64 bitlik bir blok boyutu ve değişken anahtar uzunluğu kullanan simetrik bir anahtar bloğu şifresidir. 32 bit ile 448 bit arasında değişen uzunlukta bir anahtar gereklidir. Blowfish'in 14 tur veya daha düşük varyantları vardır. Blowfish çok güvenli bir şifrelemedir ama onun yerini küçük 64 bit blok boyutu nedeniyle Twofish ve Rijndael's almıştır. Blowfish şimdiye kadar geliştirilen en hızlı blok şifrelerinden biridir. Ama yine de hızında

yavaşlama Blowfish'in bazı uygulamalarda kullanılmasını engellemiştir. Blowfish, herkesin patent ve telif hakları olmaksızın şifreleme kullanmasına izin vermek için yaratılmıştır. Blowfish bugüne kadar sadece kamusal alanda kalmıştır. Zayıf anahtar probleminden muzdarip olmasına rağmen, saldırıya karşı başarılı olduğu bilinmektedir.(Devi ve ark., 2015)

e) RC4, akış şifreleme algoritmasıdır. Aynı algoritma hem şifreleme hem de şifre çözme için kullanılır, çünkü veri akışı üretilen anahtar dizisiyle XOR yapılır. Anahtar akışı, kullanılan düz metinden tamamen bağımsızdır. 256 bitlik bir durum tablosunu başlatmak için 1 ila 256 bit arasında değişken uzunlukta bir anahtar kullanır. Durum tablosu, daha sonraki sahte rasgele bitler üretmek için kullanılır ve daha sonra şifreli metni üretmek için düz metinle XOR yapılan sahte rasgele bir akış oluşturur.(Mousa ve Hamad, 2006)

1.1.2. Steganografi

Son yıllarda, steganografi araştırmacıların dikkatini giderek daha fazla çeken bir konu haline gelmiştir. Steganografi, gizli mesajın, taşıyıcı medya dosyasında belirgin bir bozulma olmaksızın taşıyıcıya yerleştirildiği bir veri gizleme tekniğidir. Taşıyıcı medyası, metin dosyası, görüntü, video, ses vb. gibi herhangi bir dijital medya dosyası olabilir. Görüntü dosyaları mevcut diğer medyalara kıyasla taşıyıcı medyası olarak daha çok kullanılmaktadır.

Kriptografi ve damgalama (Watermarking), dijital bilgiye güvenlik sağlamak için kullanılan diğer iki kavramdır. Kriptografide, şifreleme algoritmasının şifreli çıktısı (şifreleme) orijinal mesajı çıkarmak için 3. kişilerin ilgisini çeker. Diğer taraftan, damgalamanın amacı, her iki teknik de veri gizleme düzenine sahip olmakla birlikte steganografiden farklıdır. Damgalama, gizli mesajın çıkarılmasını veya manipüle edilmesini zorlaştırırken, steganografi herhangi bir gizli mesajın varlığını gizlemeyi amaçlar.

Uygulamada, steganografik bir algoritma tasarlanırken, algılanamazlık, gömme kapasitesi ve güvenlik olmak üzere üç ana özellik dikkatle düşünülmelidir (Şekil 1.2.). Stego görüntüsünün gömme kapasitesi ve algılanamazlığı birbiriyle ters orantılıdır. Gömme kapasitesi arttıkça, stego görüntüsünün kalitesi de düşer. Bir steganografik algoritma tasarlamanın ana hedefi elde edilen stego görüntüsünü görsel ve istatistiksel olarak taşıyıcı görüntüye mümkün olduğunca yüksek gömme oranı ile benzer tutmaktır. Üçüncü özellik olan güvenlik (veya sağlamlık), bir steganografi algoritmasının gizli

mesajı kırpma, döndürme, filtreleme vb. gibi birçok görüntü işleme operasyonunu uyguladıktan sonra korumasını sağlamaktadır.

Steganografi algoritmaları genel olarak uzamsal(spatial) veya görüntü düzlemi ve frekans veya dönüşüm düzlemi olmak üzere iki kategoriye ayrılabilir. Uzamsal düzlemde, gizli mesaj bitleri, piksel değerlerini doğrudan manipüle edilerek taşıyıcı görüntüye yerleştirilir. Öte yandan frekans veya dönüşüm düzleminde, bazı matematiksel fonksiyonları kullanarak taşıyıcı resminin piksel değerlerinden hesaplanan frekans katsayılara gizli bitler yerleştirilir.

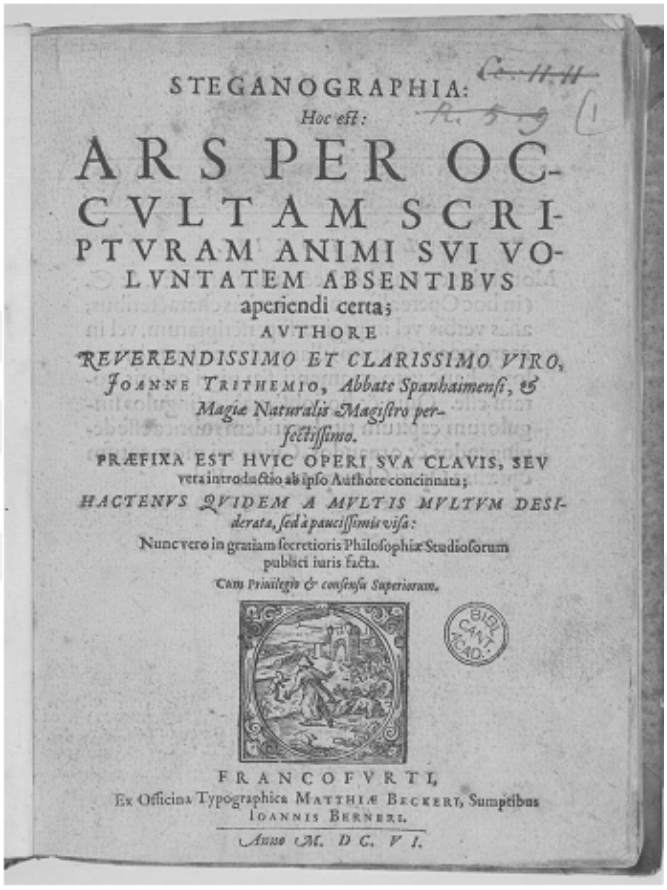
Şekil 1.3. Bir steganografi sisteminin 3 temel değerlendirme kriteri

Uzamsal düzlemde uygulanan yöntemler daha az hesaplama ve zaman karmaşıklığına sahiptir, ancak bazı saldırılara karşı nispeten daha az dayanıklıdır. Uzamsal düzlemde algoritmaları, çok iyi algılanamazlık ile çok yüksek bir gömme kapasitesine sahiptir.

1.1.2.1. Tarihte steganografi kullanımı

Steganografi kelimesi, Yunanca'da gizli anlamına gelen "steganos" kelimesi ile yazma anlamına gelen "graphia" kelimesinin birleşiminden oluşmaktadır. Başka bir deyişle, steganografi mesajın saklı olduğu gizli haberleşme sanatıdır. Steganografi terimi ilk kez Johannes Trithemuius tarafından üç ciltten oluşan "Polygraphia"da ve "Steganographia"da kullanılmıştır. İlk iki cilt, mesajları şifrelemek için kullanılan eski yöntemleri tanımlarken (kriptografi), üçüncü ciltte (1499), gizli güçler, kara büyü ve ruhlarla iletişim yöntemleri ele alınmıştır. Bu cilt, 1606 yılında Frankfurt'da yayımlandı ve 1609'da Katolik Kilisesi "libri yasağı" listesine koydu. Daha sonra akademisyenler, kitabın bir kod olduğundan şüphelenmeye başlamışlar ve gizemi çözmeyi denemişlerdir. Kitabın gizli mesajını çözme çabaları, 1996 ve 1998 yıllarında iki araştırmacı birbirinden bağımsız olarak kitapta bulunan sayısız kodlanmış gizli mesajı,

birkaç arama tablosu aracılığıyla ortaya çıkardığında başarılı bir sonuca vardı.(Kolata, 1998; Reeds, 1998) Mesajların ise oldukça sıradan olduğu ortaya çıktı. Bunlardan birincisi, “Hızlı kahverengi tilki tembel köpeğin üzerinden sıçrar.” cümlesinin Latince eşdeğeridir. Bu cümle alfabenin her harfini içeren bir cümledir. İkinci mesaj da şöyleydi: “Bu mektubun taşıyıcısı sahtekâr ve bir hırsızdır. Kendinizi ona karşı koruyun. Size bir şey yapmak istiyor.”


STEGANOGRAPHIA:
Hoc eſt:
ARS PER OC-
CVLTAM SCRI-
PTVRAM ANIMI SVI VO-
LVNTATEM ABSENTIBVS
aperiendi certa;
AVTHORE
REVERENDISSIMO ET CLARISSIMO VIRO,
JOANNE TRITHEMIO, Abbate Spanhaimenſi, &
Magiæ Naturalis Magiſtro per-
fectiſſimo.
PRÆFIXA EST HVIC OPERI SVA CLAVIS, SEV
vera introductio ab ipſo Authore concinnata;
HACTENVS QVIDEM A MVLTIS MVLTVM DESI-
derata, ſed à pauciſſimis viſa:
Nunc vero in gratiam ſecretioris Philoſophiæ Studioſorum
publici iuris facta.
Cum Priuilegio & conſenſu Superiorum.
FRANCOFVRTI,
Ex Officina Typographica MATTHIÆ BECKERI, Sumptibus
IOANNIS BERNERI.
Anno M. D C. V I.

Şekil 1.4. “Steganografi” kelimesinin mucidi olan Johannes Trithemius'un “Steganographia” adlı kitabın başlık sayfası**.**

Tarihte steganografi’nin ilk örneği, Milet tiranı Histiaeus’un Susa bölgesinde hapiste olan damadı Aristagoras’a mesaj göndermesidir. Histiaeus, kölelerinden bir tanesinin saçını kazıtır ve göndermek istediği mesajı kölenin saçına dövme şeklinde işler. Kölenin saçının tekrar uzamasını beklerler ve köleyi gönderir. Yanında şüphe çekici bir şey olmadığı için köle Aristagoras’a ulaşır ve saçını tekrar kazıtıp mesajı ulaştırır.

Herodot, Sparta'yı Yunanistan'ın Pers Kralı Xerxes tarafından yapılması planlanan istilaya karşı uyarmak için steganografiyi kullanan Demeratus'un hikâyesini de belgelemiştir. Demeratus mesajını gizlemek için, balmumu ahşap bir yazı tablasını yüzeyinden sıyırmış, mesajı tahta içine çizmiş ve ardından tableti taze bir mum katmanı ile kaplayarak normal bir boş yazı tableti gibi görünmesini sağlamıştır. Tablet, şüphe uyandırmadan Sparta'ya güvenli bir şekilde taşınmıştır.

Aeneas the Tactician, mesajların kadınların küpelerine gizlenmesi ya da gizli mesajları ulaştırmak için güvercinlerin kullanılması gibi birçok yaratıcı steganografik tekniği icat eden kişi olarak bilinir. Ayrıca, harf vuruşlarının yüksekliğini değiştirerek, küçük delikler kullanarak veya bir metindeki harfleri işaretleyerek mesajların metne gizlenmesi için bazı basit yöntemleri ortaya çıkarmıştır.(Tactician, 1990)

Metinde içerisine mesajları gizlemek, dilbilimsel steganografi veya akrostiş olarak adlandırılır. Akrostiş eskiden çok popüler bir steganografik yöntemdi. Çalışmalarına eşsiz bir "imza" yerleştirmek için bazı şairler gizli mesajları dizenin ilk harflerine veya ardışık kıtalara gömerek şiirde kodladı. En iyi bilinen örneklerden biri Giovanni Boccaccio'nun Amorosa Visione'sidir. Boccaccio, her kıtanın ilk dizesinin baş harflerine diğer şiirlerinden üçünü kodlamıştır. Dilsel steganografik şema, ilk olarak Çin'de düşünülmüş ve Cardan tarafından yeniden icat edilmiş (1501-1576) "Cardan's Grinlle"nin bir örneğidir. Gizli mesajın harfleri, metnin üzerine özel maske yerleştirerek erişilebilen rastgele bir kalıp oluşturur. Burada maske, iletişim kuran taraflar arasında paylaşılması gereken gizli bir stego anahtarının rolünü oynamaktadır. (Wilkins, 1974)

Francis Bacon modern steganografik planların bir öncüsü olan bir yöntem tanımlamıştır. Bacon, italik veya normal yazı tiplerini kullanarak, eserlerinde harflerin ikili gösterimini kodlayabileceğini fark etmiştir. Böylece taşıyıcı nesnenin beş harfi beş bit yani alfabenin bir harfini tutabilir. On altıncı yüzyıl tipografisindeki tutarsızlık nedeniyle bu yöntem pek göze çarpmamıştır. (Bacon, 1640)

Bu steganografik ilkenin modern bir versiyonu Brassil tarafından tanımlanmıştır. Metnin satırlarını bir inçin 1/300 oranında yukarı veya aşağı kaydırarak metin belgelerinde veri gizleme yöntemi tanımlamıştır. Bu tür ince değişiklikler görsel olarak algılanamamaktadır. Böylelikle, mesaj basılı veya fotokopi edilmiş belgelerden de çıkarılabilir. (J. Brassil 1994)

1857'de Brewster, aslında on dokuzuncu ve yirminci yüzyıllarda birçok savaşta kullanılan çok ustaca bir teknik önerdi (Meunier, 2000). Bu teknik, mesajı oldukça küçültüp kir lekelerini andırması ama büyüteçle mesajın okunabilmesiydi. Uygulamada

bu fikri kullanmanın önündeki teknolojik engeller, metni mikroskobik boyutlara çeken teknoloji geliştiren Fransız fotoğrafçı Dragon tarafından aşılmıştır. Bu küçük nesneler burun deliklerinde, kulaklarda veya tırnaklarının altında kolayca gizlenebilir. Birinci Dünya Savaşı'nda Almanlar kartpostalların köşelerinde gizli olan bu mikro noktalara bıçakla yarık açtılar ve nişastayla tekrar doldurdular. Modern yirminci yüzyıl mikro noktaları bir sayfaya kadar metin tutabilir ve hatta fotoğraflar içerebilirdi. Müttefikler, 1941'de mikro noktaların kullanımını keşfetti. Son zamanlarda ise önemli miktarda genetik materyali etiketlemek amacıyla DNA'ya bilgi gizlemek için mikro nokta konseptinin modern bir versiyonu önerilmiştir (Clelland ve ark., 1999; Shimanovsky ve ark., 2003). Ayrıca yine son zamanlarda, toz şeklindeki mikro noktalarlar, otomobil parçalarını tanımlamada kullanmak için önerilmiştir.

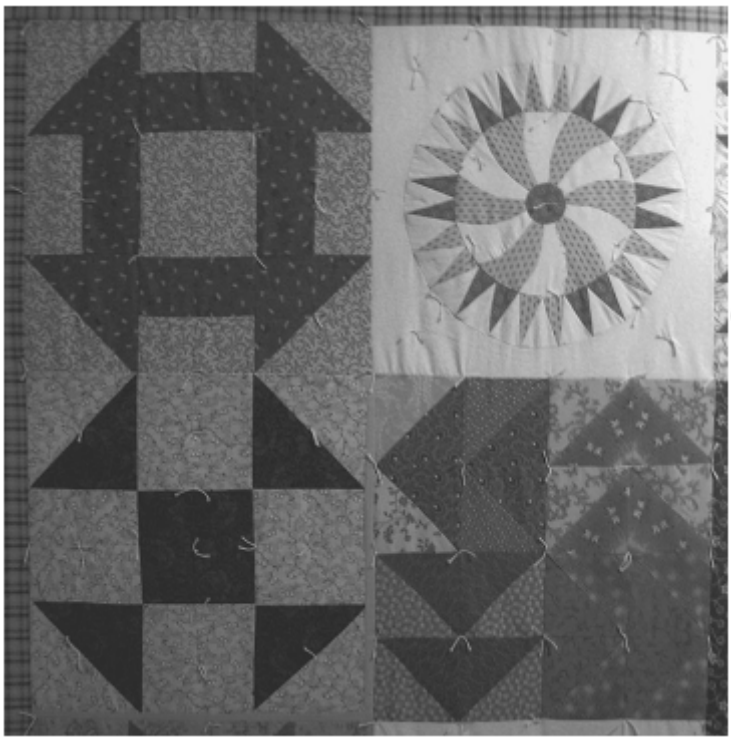
Belki de en iyi bilinen steganografi formu, görünmez mürekkeple yazı yazmaktır. İlk görünmez mürekkepler süt, idrar, sirke, seyreltilmiş bal veya şeker çözeltisi gibi organik sıvılardı. Bu mürekkeple yazılan mesajlar, kâğıt kurutulduktan sonra görünmezdi. Onları algılanması için mektubu bir mumun üzerinde ısıtılırdı. Daha sonraları, mesaj çıkarma algoritmasını ultraviyole ışığı gibi daha güvenli alternatiflerle değiştirerek daha sofistike sürümleri keşfedildi.

1966'da yaratıcı bir steganografik yöntem, bir tutsak savaşçının, Komutan Jeremiah Denton, Vietnam tutsakçılarınca TV'de röportaj yapmak zorunda bırakıldığında bir sözcüğü gizlice iletmesini sağladı. Konuştuğu gibi tutsaklarını eleştiren bir şey söyleyemeyeceğini bilerek, gözlerini Mors kodunda göz kırpıştırarak T-O-R-T-U-R-E (işkence) kelimesini söylemiştir.

Steganografi, 1978'de Viktor Korchnoi ile Anatoly Karpov arasındaki satranç Dünya Şampiyonası maçı sırasında bir tartışma konusu oldu. Maçlardan birinde Karpov'un asistanları ona yoğurtlu bir tepsi verdi. Bu, teknik olarak oyun sırasında oyuncu ile ekibi arasındaki teması yasaklayan kurallara aykırıydı. Korchnoi heyetinin başkanı Petra Leeuwerik, Karpov'un ekibinin kendisine gizli mesajlar verebileceğini savunarak hemen protesto etti. Örneğin, bir menekşe yoğurt Karpov'un beraberlik teklif etmesi gerektiği anlamına gelebilirken, dilimlenmiş bir mango oyuncuyu beraberliği reddetmesi gerektiği konusunda bilgilendirebilir. Yiyeceklerin servis süresi ek mesajlar göndermek için de kullanılabilir (zamanlama kanallarında steganografi). Soğuk Savaş sırasında satrançta egemen olan aşırı paranoyanın bir sonucu olan bu protesto oldukça ciddiye alındı. Yetkililer, oyunda sabit bir zamanda Karpov'u yalnızca bir çeşit yoğurt

(menekşe) tüketimine sınırladılar. Bu koruyucu önlemi steganografinin kullanılmasını önlemek için aktif bir koruyucu olarak yorumlayabiliriz. (Johnson, 2008)

1990'lı yıllarda, medyada Yeraltı Demiryolunda kullanılan bir "yorgan kodu" hikâyesi ortaya çıktı. Yeraltı Demiryolları, on dokuzuncu yüzyılın ilk döneminde ABD'de siyah kölelerin kölelikten kurtulmasına yardımcı olan güvenli evler gibi kendiliğinden ortaya çıkmıştır. Ozella Williams adında Güney Carolina'lı bir kadının anlattığı habere göre, yolculuklarının yönü veya atacakları sonraki adımı sözlü olarak bilgilendirmek için çitlerinde insanlara sempatik gösterilen yorganlar sergilenirmiş. Mesajlar Amerikan döşemesi yorganlarında yaygın olarak bulunan geometrik desenlerde gizlenirmiş (Şekil 1.5). Çitlerde yorganları havalandırmak ortak bir şey olduğu için, usta/patron sergilenen yorganlar konusunda şüphe duymazdı. (Jacqueline Tobin, 1999)



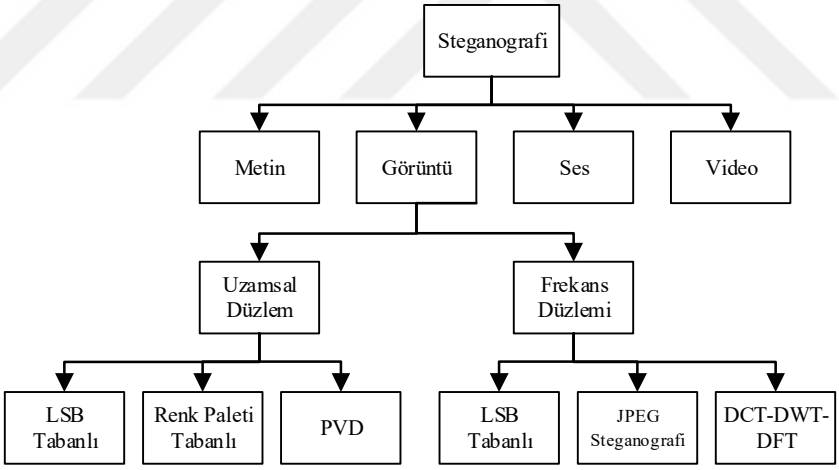
Şekil 1.5. Washington yakınlarındaki Ulusal Kriptoloji Müzesinde sergilenen bir Amerikan yamalı yorganındaki semboller

Steganografi konusundaki son patlama, dijital medyanın ani ve yaygın bir şekilde kullanılmasının yanı sıra internetin hızla genişlemesinden kaynaklanmaktadır. Resimler, video ve sesleri arkadaşlarımızla ve ailenizle paylaşmak artık yaygın bir uygulamadır. Bu tür nesneler, gizli mesajları iyi niyetle gizlemek için çok elverişli bir ortam sağlar: tipik dijital medya dosyaları, gizli bir mesajı fark edilmeyecek şekilde

kodlamak için değiştirilebilen çok sayıda bireysel örnekler içerir. Steganografi kullanmak isteyenlerin teknik uzmanlık geliştirmeye ihtiyacı yoktur, çünkü veri saklama işini internetten ücretsiz indirebileceği bir bilgisayar programı ile yapılabilir. İnternette bulunan yüzlerce steganografik ürün arasından seçim yapılabilir. Güvenlik, gizlilik ve anonimite üzerine odaklanan bazı yazılım uygulamaları resim ve müziklerdeki şifrelenmiş mesajları ek bir koruma katmanı olarak gizleme imkânı sunar. Bu tür programlara örnek olarak Steganos ve Stealthencrypt verilebilir.

1.1.2.2. Görüntü steganografi teknikleri

Görüntü steganografisi taşıyıcı medya türünün görüntü olduğu steganografi dalıdır. Steganografi alanında en çok çalışma bu medya türünde yapılmaktadır. Çünkü insanlar arasındaki gündelik iletişimde görüntü daha çok kullanıldığı için mesaj gizlenmiş bir görüntü diğer medya türlerine göre daha az dikkat çekecektir. Görüntü içerisine gizlenecek medya türü düz metin olabileceği gibi görüntü içerisine görüntü, ses ve video gibi medya türleri de gizlenebilmektedir. Bunlardan metin ve görüntü çalışmalarda en çok kullanılan mesaj dosyası türleridir.



Şekil 1.6. Taşıyıcı medya türüne göre steganografi

Görüntü steganografi teknikleri gömme yöntemine göre ikiye ayrılmaktadır. Bunlar görüntü veya uzamsal düzlem yöntemleri ve frekans düzlemi yöntemleridir. Uzamsal veya görüntü düzleminde gömme işlemi doğrudan görüntünün pikselleri mesaj bitleriyle değiştirilerek yapılmaktadır.

Frekans düzleminde ise doğrudan pikselleri değiştirmek yerine ayrık kosinüs dönüşümü, ayrık dalgacık dönüşümü gibi dönüşümler kullanılarak görüntü frekans katsayılarına dönüştürülür Gömme işlemi ise bu frekans katsayıları değiştirilerek

yapılmaktadır. Gömme işleminden sonra ters dönüşümler uygulanarak görüntü tekrar uzamsal düzleme geçirilmektedir.

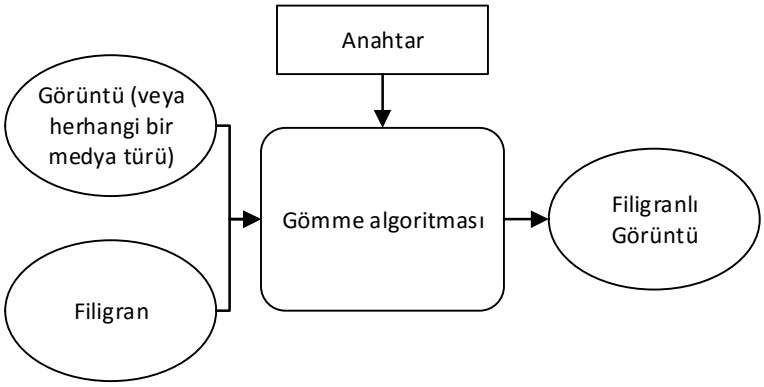
Şekil 1.6'de sıkça kullanılan uzamsal ve frekans düzlemi yöntemleri verilmiştir. Şekilde de görülebileceği gibi LSB değiştirme yöntemi iki düzlemde de ortaktır. LSB değiştirme yöntemi bu yöntemler arasında en yaygın olarak kullanılan yöntemdir. Bu nedenle LSB değiştirme yöntemine dayalı birçok farklı yöntem önerilmiştir. Bölüm 3.2'de LSB değiştirme yöntemi ayrıntılı olarak açıklanmıştır.

1.1.3. Damgalama

Son yıllarda, dijital medyanın popülerliği yaygınlaştıkça, güvenlikle ilgili konular üstün bir endişe haline gelmiştir. Dijital filigran(Watermark) ilk kez 1992 yılında Andrew Tirkel ve Charles Osborne tarafından keşfedilmiştir (Schyndel ve ark., 1994; Hemani, 2017). Filigran kelimesi Almanca bir terim olan "Wessmark"dan türetilmiştir. İlk filigranlar 13. yüzyılda İtalya'da gelişmekteydi, ancak bunların kullanımı Avrupa'ya da hızla yayıldı. Damgalama özelliği, bir mesajın bir başkasına yerleştirildiği ve iki mesajın birbiriyle ilişkili olduğu, steganografi'nin özel teknikleri olarak görülebilir. Damgalama, bir kişinin dijital ortama özel haklar bildirimlerini veya diğer doğrulama mesajlarını eklemesine olanak tanıyan filigran tekniğidir. Görüntü kimlik doğrulama, dijital görüntülerin kimliğini doğrulamak için kullanılan dijital filigran uygulamalarının biridir. Dijital filigran, ses veya görüntü verileri gibi gürültüye dayanıklı bir görüntüye gizlice yerleştirilen bir çeşit işaretleyicidir. Genellikle bu tür, resimlerin telif hakkının mülkiyetini belirlemek için kullanılır. (Mohan Durvey, 2014)

"Filigran" gizli bir görüntüye dijital bilgi gizlenmesi işlemidir ancak görüntü ile ilişkili olması gerekmez. Dijital medyada akademik mülkiyet haklarının güvenliği ve uygulanması önemli bir konu haline gelmiştir. Bu özelliği anlamanın yolu, bir dijital filigran kullanarak dijital görüntüye bir düzeyde kimlik doğrulama imzası karıştırmaktır. Resmin bozulması durumunda, gömülü verilerin piksel değerleri değişecek ve orijinal piksel değerleri ile eşleşmediğinden kolaylıkla tespit edilebilecektir. Damgalama işleminin kimlik doğrulamasında kullanılabilecek birçok uzamsal ve frekans alanı tekniği vardır. Filigran teknikleri, küçük bir özellik kümesindeki performanslarına dayanarak değerlendirilir. Filigran şemaları, uygulamanın gereklerine göre geliştirilmekte ve tüm uygulamalar bu özelliklerin her birinin tamamını gerektirmemektedir, yani filigran gereksinimleri uygulamaya bağımlıdır ve bu uygulamalar için bazı arzu edilen özellikler doğasında çelişkilidir. Dijital imza, aynı zamanda, görüntü içeriğinin güvenilirliğini ve özgünlüğünü

doğrulamak için kullanılan bir doğrulama şemasıdır. Bir dijital imza, resim içeriğinin ve görüntü özelliklerinin şifrelenmiş veya imzalı bir karma değeri olabilir.



Şekil 1.7. Damgalama akış diyagramı

1.1.3.1. Damgalamanın (Watermarking) karakteristik özellikleri

Dijital Filigranın (Watermark) sergileyebileceği bir sürü önemli karakteristikler vardır. Dijital filigranın temel karakteristik özelliklerini aşağıdaki gibi ana kategorilere ayırabiliriz.

- Sağlamlık (Robustness): Filigran (Watermark), resim kırpma, dönüştürme, sıkıştırma gibi genel görüntü işleme işlemlerine karşı dayanıklı olmalıdır.
- Algılanamazlık (Imperceptibility): Filigranlanmış görüntü orijinal görüntüyle aynı görünmelidir. Gözle bakıldığında filigran bulunduğu anlaşılmamalıdır.
- Güvenlik (Security): Yetkisiz biri, gömülü filigranı algılayamamalı, alamamalı veya değiştirememelidir.
- Şeffaflık (Transparency): Şeffaflık, insan duyusunun özellikleriyle ilgilidir. Şeffaf bir filigran yapay bir görünüme veya özellik kaybına neden olmaz.
- Kapasite (Capacity): Kapasite, kaç bilgi bitinin gömüldüğünü tanımlar. Aynı zamanda, bir belgeye paralel olarak birden fazla filigran gömme olasılığı da ele alınmaktadır. Kapasite ihtiyacı daima iki önemli gereksinime, yani algılanamazlık ve dayanıklılığa karşı etki gösterir. Daha yüksek bir kapasite genellikle dayanıklılık kuvveti veya algılama kabiliyeti veya her ikisinin de karşılığı olarak elde edilir.

1.1.3.2. Damgalamanın (Watermarking) özellikleri

Filigran sistemi çok önemli popüler özelliklere sahiptir. Bu özelliklerden bazıları şunlardır:

- Etkililik (Effectiveness): Filigranlanmış bir resimdeki mesajın doğru algılanması ihtimalidir. İdeal olarak bu olasılığın 1 olması gerekir.
- Görüntü doğruluğu (Fidelity): Damgalama, orijinal bir görüntüyü içine mesaj ekleyerek değiştiren bir işlemdir. Bu nedenle görüntü kalitesini kesinlikle etkilemektedir. Görüntü kalitesinin azalmasını asgari seviyede tutmak istenildiği için, görüntünün doğruluğunda hiçbir belirgin fark görülemez.
- Yük boyutu (Payload size): Veri yükü, orijinal görüntüye yerleştirilen bit sayısını belirtir. Görüntü kalitesini düşürmeden gizlenebilecek en yüksek miktardaki bilgidir. Orijinal verilerdeki gizli bilgilerin miktarı ile hesaplanabilir. Bu özellik, çıkarma işlemi sırasında etkili bir şekilde algılanabilmesi için ne kadar verinin filigran olarak gömülmesi gerektiğini gösterir.
- Yanlış pozitif oranı: Gerçekte filigran gömülü olmayıp filigran gömülü olarak tanımlanan dijital eser sayısıdır. Filigran sistemleri için bu çok düşük tutulmalıdır.
- Sağlamlık (Robustness): Filigranlanmış bir çalışmanın ömrü boyunca, bir bozuk kanal üzerinden iletim yoluyla veya filigranı kaldırmaya veya onu algılanılamaz hale getirmeye çalışan birkaç kötü niyetli saldırıya karşı dayanıklılığıdır. Sağlam bir filigran, ilave Gauss gürültüsü, sıkıştırma, yazdırma ve tarama, döndürme, ölçekleme, kırpma ve diğer birçok işleme dayanabilir olmalıdır.

1.1.3.3. Damgalama uygulamaları

Damgalama ile ilgili çeşitli uygulamalar bulunmaktadır. Bunlar aşağıda listelenmiştir.

a) Telif hakkı koruması

Filigranın en önemli uygulamalardan biri, yetkisiz kullanıcıya telif hakkı korumasıdır. Dijital medyanın sahibi, gömülü verileri bir kanıt olarak kullanarak bir telif hakkı anlaşmazlığı söz konusu olduğunda ortaya çıkarabilir.

b) Yayın İzleme

Bu uygulama, yetkisiz yayın istasyonlarını izlemek için kullanılır. İçeriğin gerçekten yayınlanıp yayınlanmadığını doğrulayabilir.

c) Tamper Algılama

Bozulmuş filigranlara müdahale olup olmadığının tespiti için kullanılır. Filigran bozulursa veya imha ediliyorsa, bir müdahalenin varlığını gösterir ve dolayısıyla dijital içeriğin güvenilir olamayacağı anlamına gelir.

d) Doğrulama

Filigran, görüntünün özelleştirilip özelleştirilmediğini algılamak için gömülür ve bu işlem doğrulama için kullanılabilir. Dürüstlük doğrulaması, bir görüntünün modifikasyonunda düşük sağlamlık derecesine sahip kırılgan veya yarı kırılgan filigran kullanılarak başarılabilir.

e) Parmak İzi

Parmak izinin asıl amacı müşterileri korumaktır. Biri ürünün yasal bir kopyasına sahipse, yasadışı olarak yeniden dağıtılırsa, parmak izi bunları önleyebilir. Bu, her bir alıcı için tek bir güçlü filigranı gömmek suretiyle tüm işlemi izleyerek başarılabilir.

f) İçerik Açıklama

Bir filigran, taşıyıcı resmin etiketleme ve altyazı gibi bazı ayrıntılı bilgilerini içerebilir. Bu tür bir uygulama için filigranın kapasitesi nispeten geniş olmalı ve sağlamlık birinci derece önemli olmamalıdır.

g) Tıbbi Uygulamalar

Tıbbi alanda filigran, hastanın raporu gibi hastanenin bilgilerini yetkisiz kişilerden korumak amacıyla önemlidir. Güvenlik ve doğrulama dijital verilerin internet üzerinden kolaylıkla dağıtıldığı tıbbi alanlarda giderek önem kazanmaktadır.

2. KAYNAK ARAŞTIRMASI

Chan, 2011 yılında yaptığı çalışmada, modül tabanlı yeni bir LSB steganografi tekniği önermiştir. Önerilen yöntemde ayrıca sıkıştırma da uygulamıştır. Önerilen yöntemi, 2 farklı 512x512 piksel çözünürlüğünde gri-skala görüntü üzerine 7 farklı 256x512 piksel çözünürlüğünde gri-skala görüntü gizleyerek test etmiş ve test sonucunda 31-37dB PSNR değerleri elde etmiştir. Klasik LSB yöntemine göre %7 ile %11 arasında değişen artışlar elde edilmiştir. Aynı zamanda, Thien ve Lins'in yöntemi ile Chang ve ark. önerdiği yöntemi de uygulamış, kendi yönteminde daha yüksek PSNR değeri elde ettiğini vurgulamıştır.(Chang ve ark., 2003; Thien ve Lin, 2003; Chen, 2011)

Bu çalışmada Akhtar ve arkadaşları, Chen 'in önerdiği algoritmayı geliştirerek yeni bir modül tabanlı LSB steganografi yöntemi önermişlerdir. Önerilen yöntemi, 2 farklı 512x512 piksel çözünürlüğünde gri-skala görüntü üzerine 10 farklı 256x512 piksel çözünürlüğünde gri-skala görüntü gizleyerek test etmişlerdir. Test sonucunda 32-40dB arası PSNR değerleri elde etmişlerdir. Klasik LSB yöntemine göre %3 ile %25 arasında değişen artışlar elde etmişlerdir. Aynı zamanda, Thien ve Lins'in yöntemi, Chang ve ark. önerdiği yöntemi ve Shang-Kuan Chen in önerdiği de uygulamış, kendi yöntemlerinde diğer 3 yönteme göre daha yüksek PSNR değeri elde ettiklerini söylemişlerdir.(Akhtar ve ark., 2017)

Rajput ve arkadaşları, bu çalışmada, renkli ve gri-skala görüntüler için RSA kriptografi ve STW sıkıştırma yöntemini kullanarak görüntü üzerine gizleme uygulaması yapmışlardır. MATLAB programı üzerinde 8 farklı görüntü üzerine gizleme yapılmış, fakat gizlenen veri boyutu belirtilmemiştir. Yapılan analiz sonucunda 77-83dB arası değişen PSNR değeri ve yaklaşık klasik LSB yöntemine göre %50 PSNR artışı elde etmişlerdir.(Rajput ve ark., 2016)

Chikouche 2017 yılında yaptığı çalışmada klasik LSB yöntemini, AES kriptografi yöntemi ve Deflate sıkıştırma yöntemi ile birleştirmiş ve uygulamıştır. LSB gömme işlemi sırasıyla değil, sözde-rastgele üreteç ile rastgele yapılmıştır. Önerdikleri yöntemin yanında mevcut 3 yöntemi de uygulamış ve histogramlarını incelemişlerdir. 512x512 piksel renkli görüntü üzerine 3264 bit veri gömmüşler ve kendi yöntemlerinin güvenlik kriterinde daha iyi olduğunu vurgulamışlardır.(Chikouche ve Chikouche, 2017)

Arora ve arkadaşları yapmış oldukları çalışmada AES şifreleme uyguladıkları mesajı klasik LSB yöntemiyle gizlemişler ve PSNR ve MSE kriterlerine göre incelemişlerdir. Önerdikleri bu yöntemi ve klasik LSB yöntemi 2 farklı renkli görüntüye uygulamış ve bu yöntem sayesinde güvenlik ve kapasite de iyileştirme sağladıklarını söylemişlerdir. (Arora ve ark., 2016)

Manjula ve Shivakumar yapmış oldukları çalışmada, önce AES daha sonra ECC ile şifreledikleri mesajı LZW algoritması ile sıkıştırmış ve klasik LSB yöntemi ile gömme işlemi yapmışlardır. Farklı görüntüler üzerine 32 bit mesaj gizlenmiş ve 80 dB PSNR değeri elde etmişlerdir. Daha sonra aynı görüntü üzerine 32-288 bit arası değişen uzunlukta mesaj gizlenmiş ve 77-81 dB PSNR değeri elde etmişlerdir. Ayrıca mesaj 2 d efa şifrelendiği için 2 kat güvenlik sağladıklarını belirtmişlerdir. (Manjula ve Shivakumar, 2016)

Zhou ve ark. 2016 yılında yaptıkları çalışmada klasik LSB değiştirme yöntemi ile RSA şifreleme algoritmasını birleştirmişlerdir. 256x256 piksel çözünürlüğündeki renkli görüntü üzerine 90x90 piksel çözünürlüğünde gri-skala görüntü gizlenerek test yapılmış ve klasik LSB yöntemine göre PSNR değerinde 3dB artış sağlamışlardır. (Zhou ve ark., 2016)

Sugathan, yaptığı çalışmada, klasik LSB değiştirme yöntemine dayalı yeni bir yöntem önermiştir. RGB görüntü 3'erli bloklara ayrılmış, her bloktaki kanalların son biti mesaj biti ile değiştirilerek 8 bitlik mesaj gömülmüş, 9. bit ise yön belirten bit olarak kullanılmıştır. ,son bitin 0 veya 1 olma durumuna göre gömme işlemi sağa veya sola doğru yapılmıştır. Bu yöntem 10 farklı görüntü üzerinde test edilmiş, ve PSNR değerinde artış sağlandığı görülmüştür. (Sugathan, 2016)

Islam ve arkadaşları yaptıkları çalışmada MSB(en yüksek değerlikli bit)'lerin farkını kullanan yeni bir yöntem önermişlerdir. Bu yöntemde 5. ve 6. bitlerin arasındaki fark, mesaj biti ile kıyaslanmış, 5. bit değiştirilerek farkın mesaj bitine eşit olması sağlanmıştır. Önerilen yöntem 2 gri-skala ve 2 RGB görüntüye 32 kB veri gizlenerek test edilmiş, mevcut MSB değiştirme tabanlı yöntemlere göre daha yüksek PSNR değerleri elde edilmiştir. (Islam ve ark., 2016)

Kalita ve Tuithung 2016 yılında yaptıkları çalışmada, 8nPVD tabanlı yeni bir görüntü steganografi yöntemi önermişlerdir. Bu yöntemde taşıyıcı görüntü 3x3'er piksellik bloklara ayrılmış, merkez dışındaki piksellere veri gömülmüş ve aradaki piksel farkına göre merkez piksel değiştirilmiştir. Önerilen yöntem gri-skala görüntüye uygulanmıştır. Wu et al ve M. Khodaei ve A. Feaz nin önerdiği yöntemlerle kıyaslama

yapılmış, kapasitede artış sağlandığı vurgulanmıştır.(Wu ve ark., 2005; Khodaei ve Faez, 2012; Kalita ve Tuithung, 2016)

RSA kriptoloji algoritması ile 1-2-4 LSB algoritması birleştirilerek yeni bir yöntem önerilmiştir. Gömme işleminden önce mesaj RSA algoritması ile şifrelenmiş, daha sonra taşıyıcı görüntünün kenarları tespit edilmiş ve 1-2-4 LSB algoritması ile bu piksellere gömme işlemi uygulanmıştır. 4 farklı RGB ve 3 farklı gri-skala görüntü üzerinde uygulanmış ve 43-48dB arası PSNR değerleri elde edilmiştir. Gizlenen veri miktarı hakkında bilgi verilmemiştir.(Goyal ve ark., 2015)

Nickfarjam ve arkadaşları yaptıkları çalışmada parçacık sürü optimizasyonu (PSO) ve komşu piksellere bakılarak LSB gömme işlemi uygulamışlardır. Gömme işleminde, komşu piksellerden 3 özellik çıkarılmış ve bu özellikleri kullanarak PSO ile en uygun piksele gömme işlemi yapılmıştır. Bu yöntem 256x256 piksel çözünürlüğünde gri-skala 4 taşıyıcı görüntü içerisine 128x128 piksel çözünürlüğünde görüntü gizleyerek test edilmiştir. Elde edilen sonuçlar, klasik LSB, Chang ve ark., Khodaei ve ark. ve LSB-PSO yöntemleriyle kıyaslanmıştır. Bu yöntemlere göre önerilen yöntemde daha yüksek PSNR değeri elde edilmiştir. %1 ile % 15 arasında PSNR artışı olmuştur. (Chang ve ark., 2003; Khodaei ve Faez, 2010; Bedi ve ark., 2011; Nickfarjam ve ark., 2014)

Karakış ve Güler yaptıkları çalışmada bulanık mantık ile LSB değiştirme yöntemini birlikte uygulanmıştır. Öncelikle gizlenecek mesaj bitleri 128 bitlik bir anahtar ile XOR'lanarak şifrelenmiştir. Taşıyıcı görüntüde komşu pikseller arasındaki farklara göre veri gömmek için en uygun pikseli belirleyen bulanık tabanlı sistem geliştirilmiş ve şifrelenen mesaj bitleri bu uygun piksellerin LSB'lerine gömülmüştür. Bu yöntemi ise 512x512 piksel çözünürlüğünde RGB görüntülere 15 kB veri gizleyerek test etmişler ve 69-69.9 dB arası değişen PSNR değerleri elde etmişlerdir. (Karakış ve Güler, 2014)

Bu çalışmada, klasik LSB yönteminde ki gibi RGB kanallarına eşit sayıda bit gömmek yerine, 2-2-4 şeklinde gömmek önerilmiştir. Yani Kırmızı ve yeşil kanala 2 bit, mavi kanala 4 bit gömülmüştür. Önerilen bu yöntem 2 taşıyıcı görüntü üzerinde test edilmiş ve klasik LSB yöntemine göre PSNR değerinde artış sağlandığı söylenmiştir. Ayrıca, 1-3-4 yöntemine göre yine PSNR değerinde artış sağlandığı vurgulanmıştır.(Singh ve Singh, 2015)

Akhtar ve arkadaşları 204 yılında yaptıkları çalışmada, klasik LSB metoduna dayalı iki farklı yöntem önerilmiştir. Birinci yöntemde, mesaj bitleri taşıyıcı

görüntülerin son bitlerine gömülmüş, gömme işlemi sonucu değişen bitler terslenmiştir. Gizlenen mesajı doğru çıkarabilmek için, hangi bitlerin değiştiği görüntünün içerisinde tutulmuştur. İkinci yöntemde ise, taşıyıcı görüntünün alıcı tarafta bulunduğu varsayılmış, bu nedenle hangi bitlerin değiştiğini görüntü içerisine gizlemeye gerek kalmamıştır. İki yöntem 1024x1024 piksel gri-skala 3 taşıyıcı üzerine 256x256 piksel gri-skala 6 görüntü gizlenerek test edilmiştir. Birinci yöntem ile klasik LSB yöntemine göre PSNR değerinde %0.2 ile %6 arasında, ikinci yöntem ile %0.3 ile %15 arasında artış sağlanmıştır.(Akhtar ve ark., 2014)

Bedwal ve Kumar yaptıkları çalışmada RGB-kutu haritalamasına dayalı yeni bir görüntü steganografi yöntemi önermişlerdir. Önerilen yöntem, farklı çözünürlükte 4 görüntü üzerine farklı çözünürlükte 4 görüntü gizlenerek test edilmiştir. Önerilen yöntem ile PSNR değeri 40-41dB arasında çıkmıştır. Ayrıca Amitava Nag's önerdiği yönteme göre PSNR değerinde yaklaşık %20 artış sağlanmıştır.(Nag ve ark., 2012; Bedwal ve Kumar, 2013)

Darabkh ve arkadaşları Bu çalışmada, LSB değiştirme yöntemine dayalı yeni bir yöntem önerilmiştir. Bu yönteme göre, mesaj, taşıyıcı görüntünün orasına geometrik şekle benzeyecek şekilde gömülmektedir. Bu geometrik şekli gizlenecek mesaj boyutuna göre değişmektedir. Önerilen yöntem, 512x512 piksel çözünürlüğünde 5 farklı resim üzerinde farklı boyutlarda veri gömülerek test edilmiştir. Sonuçlar ise, önerilen yönteme yakın olduğu için PVD yöntemi ile karşılaştırılmıştır. Karşılaştırma sonucunda bütün veri boyutlarında PSNR değerinde yaklaşık %3 artış sağlanmıştır.(Darabkh ve ark., 2014)

Bu çalışmada, 3 farklı görüntü, birleştirilip metin olarak kodlanmış ve taşıyıcı görüntünün R G ve B kanalına ayrı ayrı gizlenmiştir. Bu yöntem farklı görüntüler üzerinde test edilmiş ve 81 ile 86 dB arası değişen PSNR değerleri elde edilmiştir.(Dongardive ve ark., 2015)

Bu çalışmada, klasik LSB steganografisi ile RC4 şifreleme algoritması birleştirilerek güvenlik ilkesinde iyileştirme sağlanmıştır. Bu yöntem 2 taşıyıcı görüntü üzerine 6 farklı mesaj görüntüsü gizlenerek test edilmiş, 54-56 dB arası değişen PSNR değerleri elde edilmiştir.(Akhtar ve ark., 2013)

Bu çalışmada, AES ve LSB değiştirme yöntemine dayalı yeni bir yöntem önerilmiştir. Öncelikle mesaj, AES algoritması ile şifrelenmiştir. Daha sonra, sırasıyla her pikselin R, G ve B kanallarının MSB'lerine bakılmış, MSB'lerindeki 0 ve 1 sayısına göre mesaj bitleri son bitlere gizlenmiştir. Önerilen yöntem, 8 farklı taşıyıcı görüntü

üzerinde denenmiş ve 56 ile 76 dB arasında değişen PSNR değerleri elde edilmiştir. Ayrıca önerilen yöntem mevcut 3 farklı yöntemle de karşılaştırılmış ve önerilen yöntemin daha yüksek PSNR değerine sahip olduğu vurgulanmıştır.(Islam ve ark., 2014)

3. MATERYAL VE YÖNTEM

3.1. Materyal

Bu bölümde tezde kullanılan mevcut yöntemlerin ve önerilen yöntemin ayrıntılarını içermektedir. Mevcut ve önerilen yöntemlerin kodları MATLAB programı kullanarak yazılmış, daha sonra bu kodlar MATLAB GUI ile arayüz haline getirilmiştir. Kodların ve arayüzün tasarımı ve analizi için 2.6GHZ Intel® Core ™ i7 CPU ve 16GB RAM'e sahip Windows 10 yüklü bilgisayar kullanılmıştır.

3.2. LSB Steganografi Yöntemi

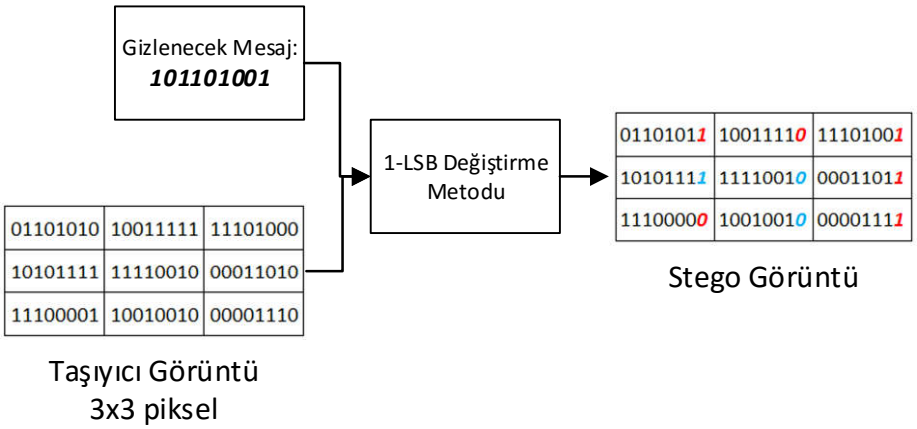
LSB (En Düşük Değerlikli Bit) değiştirme yöntemi, steganografi'de en çok kullanılan ve uygulanması en basit yöntemlerden bir tanesidir. Bütün medya türlerinde kullanılabilmesi, LSB değiştirme yöntemini en temel yöntemlerden bir tanesi yapmaktadır.

LSB değiştirme yönteminin temel prensibi, taşıyıcı resmin sırasıyla her pikselinin en düşük değerlikli bitini mesaj biti ile yer değiştirmektir. RGB veya gri görüntüye uygulanabilir. RGB görüntüde, ister kırmızı, yeşil veya mavi kanallardan bir tanesine, ister üç kanala da gömme işlemi uygulanabilmektedir.

Aşağıda tek renkli(gri) görüntüye LSB değiştirme yöntemini uygulama adımları verilmiştir:

1) Taşıyıcı görüntüyü al.
2) Gömülecek mesajı al ve mesajı bitlerine ayır.
3) Taşıyıcı görüntünün ilk pikselinden başlayarak sırasıyla en düşük değerlikli bitleri mesaj bitleri ile değiştir.
4) Gömülecek mesaj bitinceye kadar 3. adımı tekrar et.
5) Mesajı gömme işlemi bitince, sonlandırma karakteri ekle.
6) İşlemi sona erdir.

Şekil 3.1'de 1-LSB değiştirme örneğinin basit bir örneği verilmiştir. 3x3 pikselden oluşan taşıyıcı görüntüye sırasıyla 9 bitten oluşan mesaj 1-LSB yöntemiyle gömülmüştür. Gömme işlemi sonucunda oluşan stego görüntüyü incelediğimiz zaman, 9 pikselin 6 tanesinin son biti orijinal görüntüye göre değişmiştir (kırmızı renkle gösterilmiştir). Diğer 3 biti ise taşıyıcı görüntü ile aynı kalmıştır. 8'er bit, yani 0-255 arası sayılardan oluşan her bir pikselin değeri, ya 1 artmış, ya 1 azalmış ya da değişmemiştir. Görüntünün piksellerinde ki ±1'lik değişimin görüntü üzerinde büyük değişim yaratmayacağı aşikârdır.

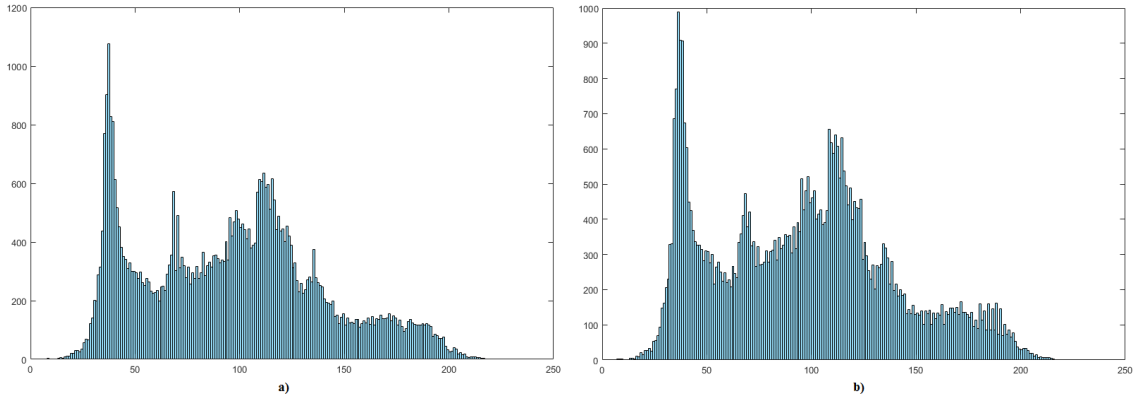


Şekil 3.1. 1-LSB değiştirme örneği

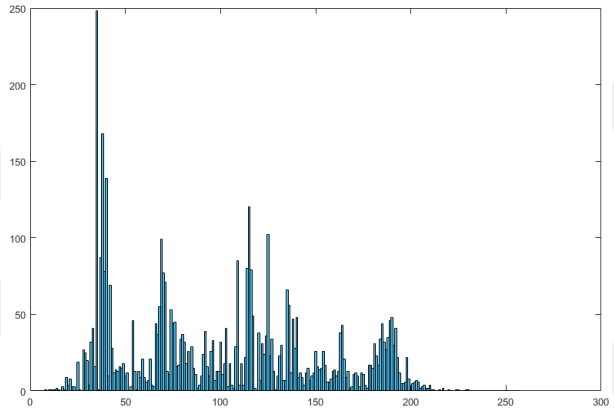
Şekil 3.2 a'da verilen 225x225 pikselden oluşan RGB görüntüye 1-LSB değiştirme yöntemi ile tam kapasitede (151875 bit, 148.3 byte) mesaj gizlenmiştir. Gizleme işlemi sonucunda oluşan stego görüntü Şekil 3.2 b'de verilmiştir. İki görüntü incelendiğinde gözle görülür bir deformasyon olmamıştır.

Şekil 3.2. a)Taşıyıcı görüntü b)Mesaj gizlenmiş stego görüntü

Şekil 3.2'de verilen taşıyıcı ve stego görüntülerin gri-skala histogramları Şekil 3.3'de, iki görüntü arasındaki histogram farkı ise Şekil 3.4'de verilmiştir. Histogramlar incelendiğinde ortaya çıkan fark gözle ayırt edilemeyecek kadar azdır.



Şekil 3.3. a)Taşıyıcı resmin histogramı b) Stego resmin histogramı



Şekil 3.4. Histogramlar arasındaki fark

Stego görüntüye gömme işleminin tersi uygulanarak gizlenmiş mesaj çıkarılabilir. Aşağıda tek renkli(gri) stego görüntüden 1-LSB değiştirme yöntemi ile gizlenmiş mesajı çıkarma adımları verilmiştir:

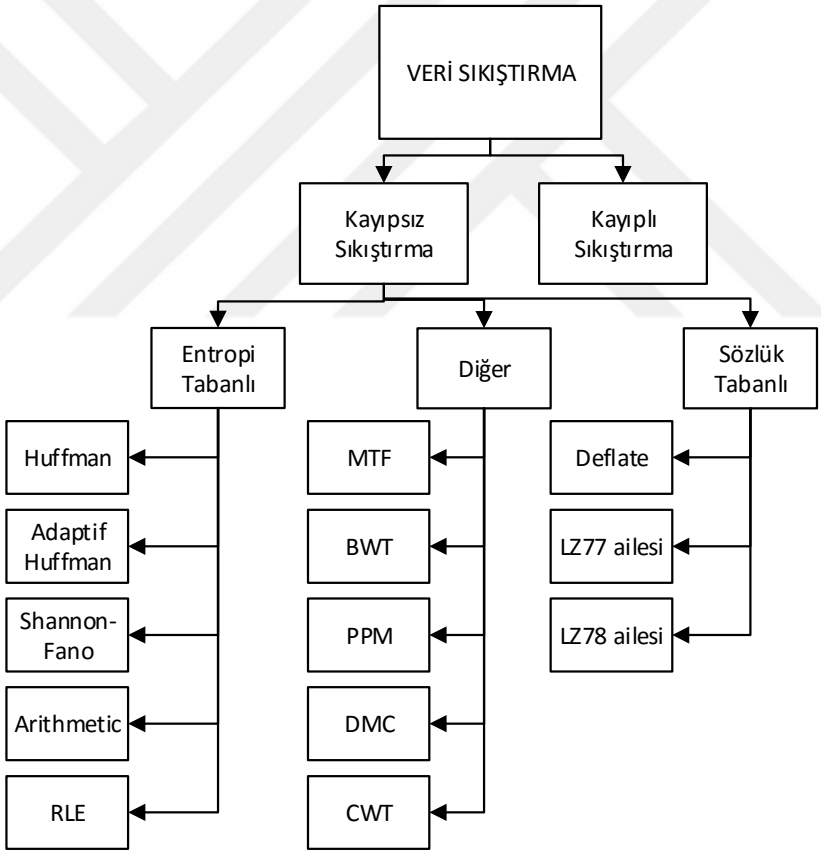
1) Stego görüntüyü al.
2) Stego görüntünün ilk pikselinden başlayarak sırasıyla en düşük değerlikli bitleri al ve yeni bir diziye sırasıyla at.
3) Sonlandırma karakteri gelinceye kadar 2. adımı tekrar et.
4) Çıkarılan bitleri birleştirerek mesajı elde et.
5) İşlemi sona erdir.

3.3. Veri Sıkıştırma Yöntemleri

Veri sıkıştırma, verileri depolamak için gereken alanı azaltmak veya verilerin iletilmesi için gereken süreyi azaltmak amacıyla veri sayısını azaltma işlemidir. Veri sıkıştırmanın amacı, orijinal veriyi yeniden oluşturacak temel özellikleri bozmadan, bir veriyi mümkün olduğunca az sayıda bit ile dijital biçimde temsil etmektir. Özellikle

günümüzde, bilgisayar ve internetin yüksek oranda kullanımı nedeniyle, bilgisayarda saklanan veya internet üzerinden gönderilen dosyaları sıkıştırmak kaçınılmaz bir zorunluluk haline gelmiştir.

Veri sıkıştırma yöntemleri sıkıştırma formatlarına göre kayıplı sıkıştırma ve kayıpsız sıkıştırma olmak üzere iki kategoriye ayrılır. Veriyi sıkıştırdıktan sonra, orijinal veri hiçbir değişikliğe uğramadan geri çıkarılabiliyorsa bu tip sıkıştırmaya kayıpsız sıkıştırma denir. Kayıpsız sıkıştırma yöntemleri, orijinal verinin aynen korunmasının önemli olduğu, herhangi bir detayın kaybedilmek istenmediği durumlarda kullanılır. Bu tür önemli verilere, tıbbî resimler, yasal nedenlerle korunan metin ve resimler, bilgisayarda kullanılan '.exe' dosyaları örnek verilebilir.(Shanmugasundaram ve Lourdusamy, 2011)



Şekil 3.5. Kayıpsız veri sıkıştırma yöntemleri

Sıkıştırma algoritmalarının diğer kategorisi olan kayıplı sıkıştırmada, orijinal veri geri çıkarma işleminden sonra aynen elde edilemez. Çünkü bu algoritmalar verilerin bazı bölümlerini geri döndürülemez bir şekilde kaldırır ve orijinal verilerin

yalnızca bir kısmı yeniden oluşturulabilir. Kayıplı sıkıştırma algoritmaları daha yüksek sıkıştırma oranına sahiptir. Bu nedenle daha çok tercih edilmektedir. Bununla birlikte, genellikle görsel kalite ve hesaplama karmaşıklığı arasında iyi bir denge kurmayı gerektirir. Görüntü, video ve ses gibi medya türleri, insan görsel ve işitme sistemlerinin ufak değişiklikleri fark edememesi nedeniyle kayıplı sıkıştırma tekniklerine daha uygundur. Metin dosyası gibi en ufak değişikliğin fark edilebileceği dosya türlerinde ise kayıpsız sıkıştırma yöntemleri tercih edilmektedir.

Şekil 3.5'de kayıpsız sıkıştırma yöntemleri verilmiştir. Bu tezde, mesajı gizlemeden önce sıkıştırmak için LZW, Arithmetic, MTF ve Deflate algoritmaları seçilmiştir. Bu algoritmalar hakkında ayrıntılı bilgi aşağıda verilmiştir.

3.3.1. Lempel-Ziv-Welch

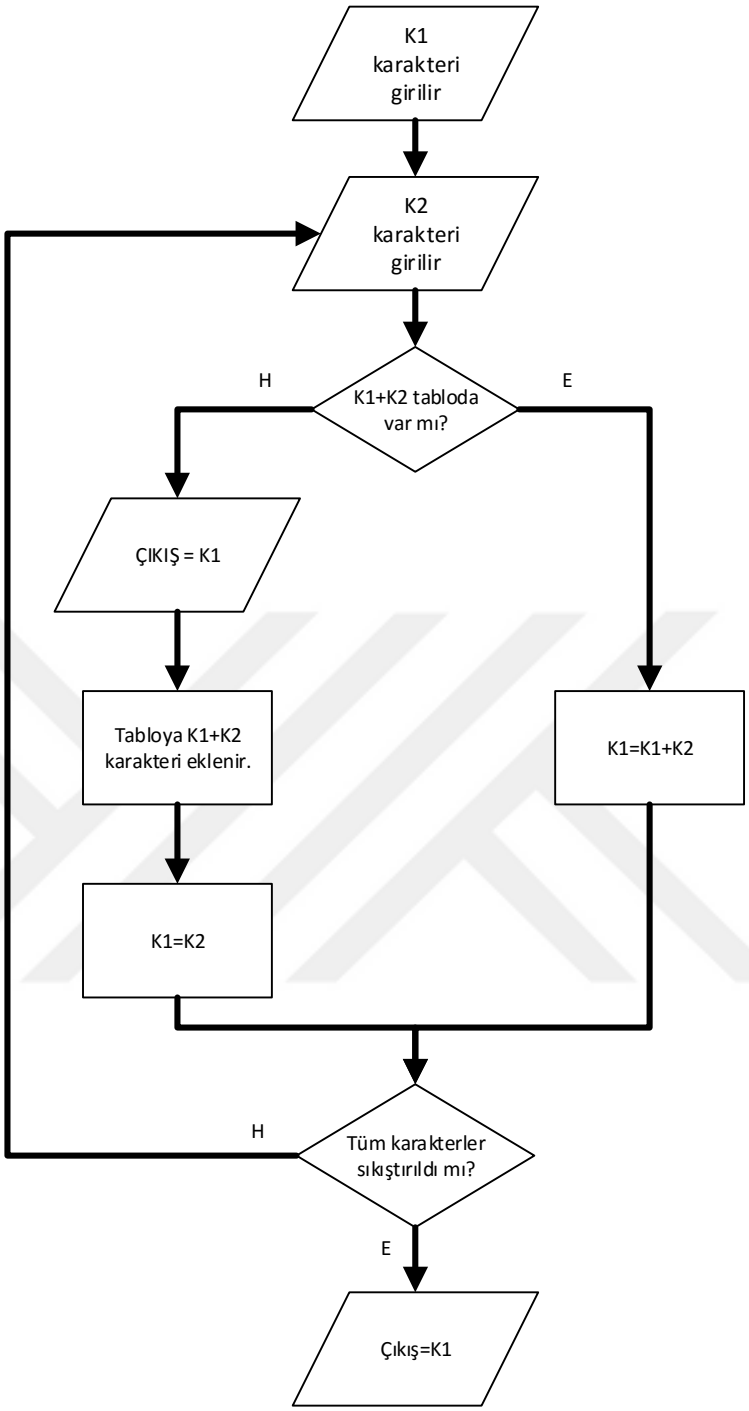
Lempel-Ziv Algoritması kayıpsız veri sıkıştırma yöntemlerinden biridir. Aslında tek bir algoritma değil, bir algoritma ailesidir. 1977 ve 1978'de Jacob Ziv ve Abraham Lempel'in yaptıkları iki çalışma sonucunda ortaya LZ77 ve LZ78 algoritmaları çıkmıştır. Bu ailedeki bütün algoritmalar ise bu iki algoritmadan türetilmiştir. (Shanmugasundaram ve Lourdusamy, 2011)

Lempel-Ziv-Welch (LZW) algoritması LZ78 algoritmasından türetilen bir sıkıştırma yöntemidir. 1984 yılında Terry A. Welch tarafından bulunmuş ve "A Technique for High-Performance Data Compression (Yüksek Performanslı Veri Sıkıştırma Tekniği)" isimli makalesinde tanıtılmıştır.(Welch, 1984)

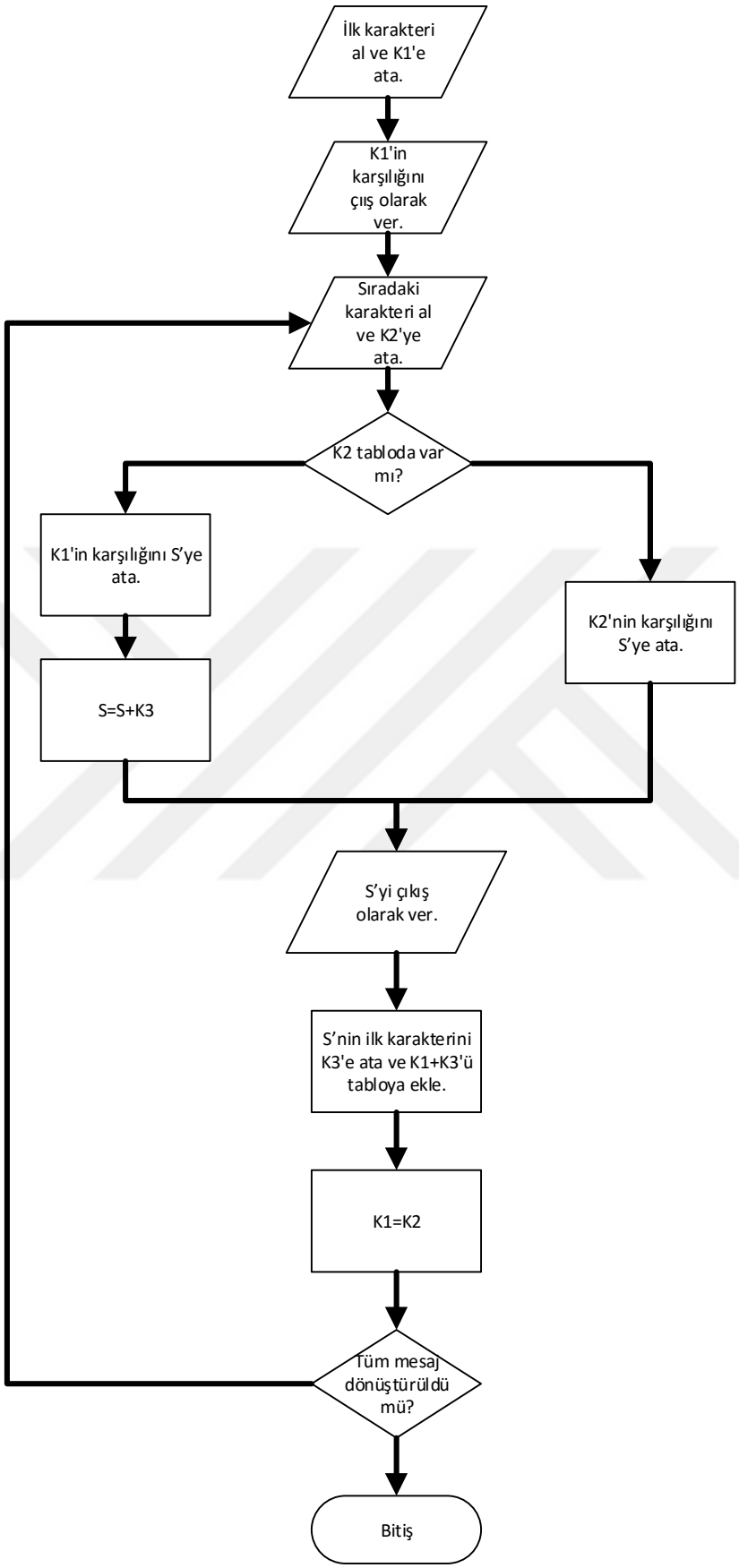
LZW, çeşitli dosya formatlarının önemli bir parçasıdır. "Gif", "tif" gibi grafik formatları, entropi kodlaması için LZW kullanmaktadır.

LZW sıkıştırma yönteminde, hazır bir sözlük bulunmamaktadır. Sözlük, sıkıştırılacak metne göre dinamik olarak oluşturulur. Bu nedenle, steganografi'de gizlenecek mesajı sıkıştırmak için LZW yöntemi kullanıldığında, alıcı tarafa sözlük veya başka bir şey iletmeye gerek yoktur. Alıcı taraf, sıkıştırılmış mesajı görüntüden çıkardıktan sonra sözlük dinamik olarak oluşturulup orijinal mesaj elde edilecektir.

Şekil 3.6'da LZW algoritmasının sıkıştırma akış diyagramı, Şekil 3.7'de çözme akış diyagramı verilmiştir.



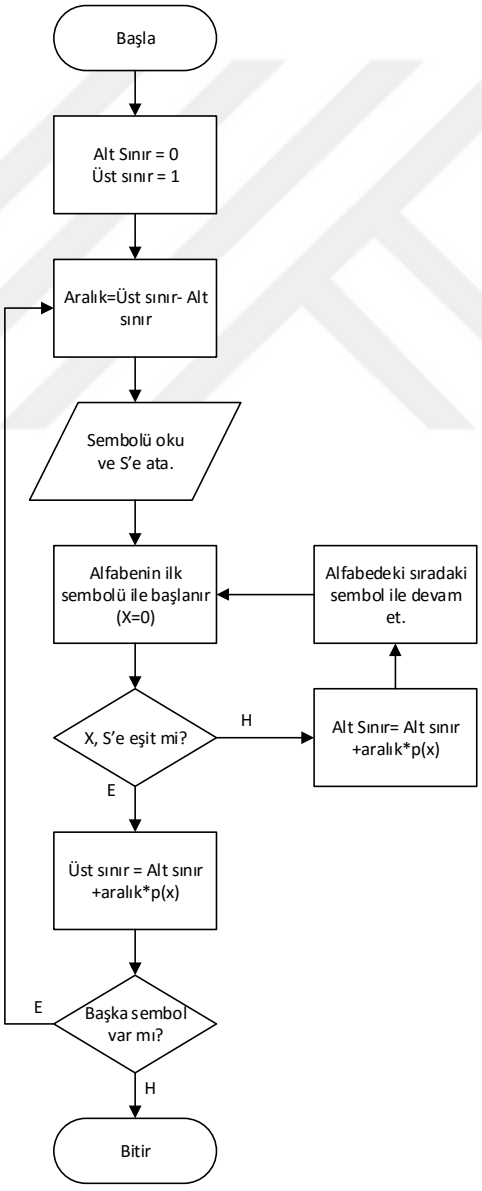
Şekil 3.6. LZW algoritmasının sıkıştırma adımları



Şekil 3.7.LZW algoritmasının çözme adımları

3.3.2. Arithmetic

Aritmetik (Arithmetic) kodlama kayıplı ve kayıpsız veri sıkıştırmada ortak olarak kullanılan bir algoritmadır. Sık karşılaşılan karakterlerin daha az bit, az karşılaşılan bitlerin ise daha çok bit ile kodlandığı entropi tabanlı bir tekniktir. Aritmetik kodlama, her giriş simgesini bir kod sözcüğüyle değiştirme fikrini tamamen ortadan kaldırır. Bunun yerine, her giriş simgesini tek bir noktalı sayı ile değiştirir. Aritmetik kodlamanın temel kavramı Elias tarafından 1960'ların başında önerilmiş ve daha sonra Pasco, Rissanen ve Langdon tarafından geliştirilmiştir. (Pasco, 1976; Rissanen ve Langdon, 1979; Langdon, 1984; Shanmugasundaram ve Lourdusamy, 2011)



Şekil 3.8. Aritmetik kodlama akış diyagramı

Aritmetik kodlamanın temel amacı, her karaktere bir aralık atamaktır. Ardından, bu aralığa bir ondalık sayı atanır. Algoritma, 0 ve 1 aralığıyla başlar. Giriş verisinde ki her bir karakter okunduktan sonra aralık, giriş karakterinin olasılığına oranla daha küçük bir aralık olarak alt bölümlere ayrılır. Bu alt-aralık yeni aralık haline gelir ve o karakterin olasılığına göre bölümlere ayrılır. Bu, her giriş karakteri için tekrarlanır. Bu işlem tamamlandığında, son aralıktaki herhangi bir noktalı sayı giriş verisini benzersiz şekilde temsil eder. Şekil 3.8'de aritmetik kodlama algoritmasının akış diyagramı verilmiştir. Burada S, orijinal mesajdaki her bir karakteri, X alfabedeki sırasını, p(x) ise o karakterin görülme olasılığını ifade eder.

3.3.3. MTF

Öne-Taşı (Move-to-Front) yöntemi entropi tabanlı veri sıkıştırma tekniklerinin verimini artırmak için tasarlanmış bir yöntemdir. Burrows-Wheeler, Huffman gibi veri sıkıştırma tekniklerinde ön işlem olarak uygulanmaktadır.

MTF yöntemi, mesajın ilk karakterinden başlayarak, alfabedeki karşılığını kod olarak döndürür ve aynı zamanda o karakteri alfabenin ilk karakteri yapar. Bu yüzden aynı karakterler mesaj içinde birbirine yakın yerleştirilirse daha küçük tam sayı değerleriyle kodlanacak, sıkıştırma oranı daha yüksek olacaktır.

Mesaj	Kod	Alfabe
steganography	**19**	abcdefghijklmnopqrstuvwxyz
st**e**ganography	19-**20**	**s**abcdefghijklmnopqrtuvwxyz
ste**g**anography	19-20-**7**	**t**sabcdefghijklmnopqruvwxyz
steg**a**nography	19-20-7-**9**	**e**tsabcdfghijklmnopqruvwxyz
stega**n**ography	19-20-7-9-**5**	**g**etsabcdfhijklmnopqruvwxyz
stegan**o**graphy	19-20-7-9-5-**16**	**a**getsbcdfhijklmnopqruvwxyz
steganog**r**aphy	19-20-7-9-5-16-**17**	**n**agetsbcdfhijklmopqruvwxyz
steganogr**a**phy	19-20-7-9-5-16-17-**4**	**o**nagetsbcdfhijklmpqruvwxyz
steganogra**p**hy	19-20-7-9-5-16-17-4-**20**	**g**onaetsbcdfhijklmpqruvwxyz
steganograp**h**y	19-20-7-9-5-16-17-4-20-**5**	**r**gonaetsbcdfhijklmpquvwxyz
steganograph**y**	19-20-7-9-5-16-17-4-20-5-**19**	**a**rgonetsbcdfhijklmpquvwxyz
	19-20-7-9-5-16-17-4-20-5-19-**14**	**p**argonetsbcdfhijklmquvwxyz
	19-20-7-9-5-16-17-4-20-5-19-14-**25**	**h**pargonetsbcdfijklmquvwxyz

Şekil 3.9. MTF sıkıştırma örneği

Şekil 3.9'da MTF algoritmasının basit bir örneği verilmiştir. "steganography" kelimesi standart İngiliz alfabesi kullanarak ilk harfinden itibaren alfabedeki sırasıyla kodlanmıştır. Kodlamadan sonra kodlanan harf alfabenin en başına getirilmiştir ve alfabedeki her karakterin sırası güncellenmiştir. "steganography" kelimesindeki bütün karakterler kodlanıncaya kadar bu işlem sürdürülmüştür. Bu örnekte sıkıştırma olmadan her bir karakterin 8 bit ile ifade edilmesi gerekirken, sıkıştırma işleminden sonra 5 bit ile ifade edilebilecek hale gelmiştir. Ayrıca sık kullanılan karakterler daha küçük değerli tam sayılarla kodlandığı için, kodlanmış mesaj başka bir sıkıştırma algoritması ile sıkıştırıldığında sıkıştırma oranı daha da artacaktır. Aynı zamanda, steganografi'de, mesaj gizlenmeden önce MTF yöntemi ile sıkıştırırsa, standart alfabe kullanıldığı için, alfabeyi alıcı tarafa iletme zorunluluğu da bulunmamaktadır.

3.3.4. Deflate

Deflate, aslında Zip ve Gzip gibi iyi bilinen algoritmalarda kullanılan popüler bir sıkıştırma yöntemidir. Deflate yöntemi, PNG görüntü, http protokolü ve PDF gibi birçok önemli program tarafından kullanılmaktadır. Deflate yöntemi, LZ77 ve Huffman kodlamasına dayalı sözlük tabanlı bir sıkıştırma tekniğidir. Üç farklı tipi bulunmaktadır. Birinci modda, giriş sembolleri sıkıştırma olmadan alt dizilere bölünür. Bu mod, sıkıştırılamayan dosyalar için veya bir dosyayı sıkıştırma olmadan bölümlendirmek istendiğinde kullanılır. İkinci mod, bir tek geçişli sıkıştırma çözümüdür. Bu modda, kodlama işlemi sırasında önceden belirlenmiş bir kodlama tablosu kullanılır. Bu mod gerçek zamanlı uygulamalarda kullanılmaktadır. Deflate'in üçüncü modu, giriş dosyasının istatistiksel özelliklerine göre üretilen sözlüklere dayalı çalışan iki geçişli bir sıkıştırma çözümüdür. Bu mod uzun gecikme nedeniyle gerçek zamanlı uygulamalar için uygun değildir.(Tahghighi ve ark., 2010)

Deflate algoritmasının ikinci modundaki en büyük dezavantajlarından biri, kodlama işlemi sırasında değişmeden kalan sabit kodlama tablolarını kullanmasıdır. Önceden belirlenmiş bir tabloya dayalı kodlama kodlama gecikmesini azaltırken, adaptif kod tablolarının sıkıştırma oranı açısından daha iyi bir performans gösterilebilir.

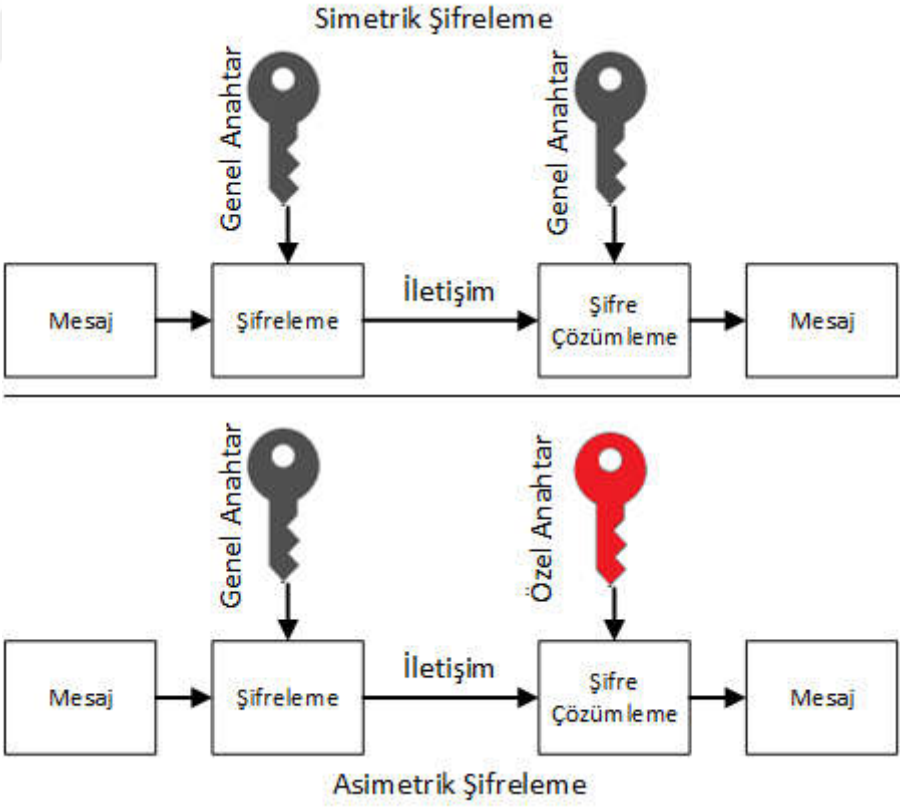
3.4. Veri Şifreleme Yöntemleri

Veri şifreleme ya da diğer adıyla kriptografi, bilgi güvenliği konusunda önemli yeri olan bir bilim dalıdır. Veri şifreleme, orijinal bilgiyi anlaşılamaz hale getirerek, 3. kişiler tarafından mesajın çözülememesini amaçlamaktadır. İnternet üzerinden insanlar arasındaki iletişimin arttığı günümüzde, kriptografi kişilerin özel verilerini korumak için

birçok farklı sistemde kullanılmaktadır. Kriptografi yöntemleri, simetrik ve asimetrik olmak üzere iki ana kategoriye ayrılmaktadır.

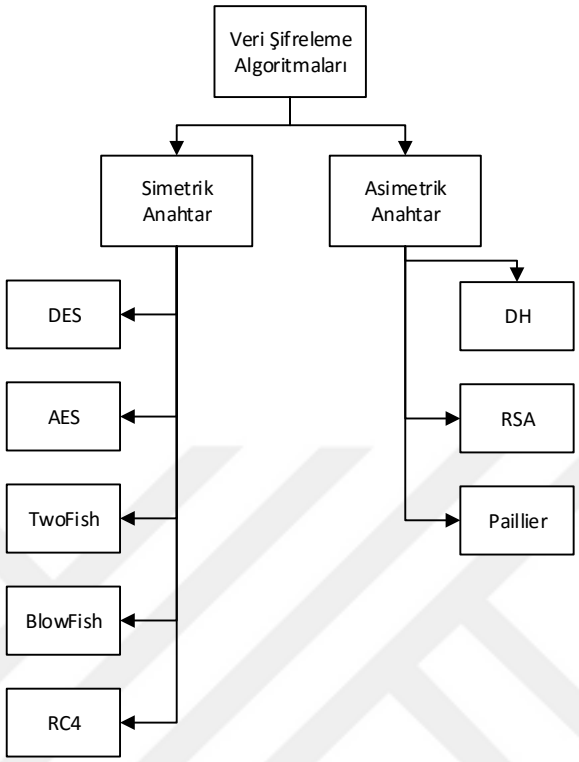
Simetrik şifrelemede hem şifrelerken hem de şifre çözerken aynı anahtar kullanılmaktadır. Bu nedenle anahtarın gizliğini sağlamak zordur ve eğer bu anahtar 3. kişilerin eline geçerse gizli mesaja rahatlıkla ulaşabilecektir. Simetrik şifreleme algoritmaları blok şifreleme ve dizi şifreleme olarak ikiye ayrılmaktadır. Blok şifrelemede veriler bloklar haline getirilerek şifrelenirken dizi şifrelemede veriler bit dizisi şekline dönüştürülüp şifrelenmektedir.

Asimetrik şifrelemede ise şifrelerken herkesle paylaşılan genel anahtar kullanılır, şifre çözme işleminde ise sadece alıcı tarafın bildiği özel anahtar kullanılmaktadır. Özel anahtar kişiye özel olduğu için, simetrik şifreleme olan anahtar dağıtma problemi asimetrik şifrelemede yoktur. Asimetrik şifrelemede anahtarlar, 2 asal sayıdan türetilir. Bu asal sayıların değeri ne kadar büyükse, şifreleme güvenliği de o kadar fazla olacaktır. Fakat bu sayılar büyüdükçe, yapılması gereken hesaplama miktarı da artacağı için süre ve donanım problemleri ortaya çıkmaktadır. Bu nedenle asimetrik şifrelemenin simetrik şifrelemeye göre şifreleme ve şifre çözme süresi çok daha uzundur.



Şekil 3.10. Simetrik ve asimetrik şifreleme

Şekil 3.10'da simetrik ve asimetrik şifreleme diyagramı, Şekil 3.11'de ise sık kullanılan simetrik ve asimetrik şifreleme algoritmaları verilmiştir.



Şekil 3.11. Sık kullanılan şifreleme algoritmaları

3.4.1. RSA

RSA şifreleme algoritması, Ron Rivest, Adi Shamir ve Leonard Adleman tarafından 1977 yılında önerilmiştir. RSA'nın açılımı, geliştiricilerinin isimlerinin baş harflerinden oluşmaktadır. RSA algoritması asimetrik şifreleme yöntemlerinden bir tanesidir. Şifreleme ve şifre çözme işlemlerinde iki farklı anahtar kullanılmaktadır fakat bu iki anahtar birbiri ile ilişkilidir. Şifreleme için kullanılan anahtar genel anahtardır ve herkes tarafından bilinir. Şifre çözme için kullanılan anahtar ise özel anahtardır ve sadece alıcı tarafta bulunur.

RSA algoritmasının şifreleme adımları aşağıdaki gibidir:

1) p ve q gibi iki asal sayı giriş parametresi olarak alınır.
2) $n=p*q$ taban değeri ve $\varphi(n)=(p-1)*(q-1)$ değeri hesaplanır.
3) $1 < e < \varphi(n)$ ve $\varphi(n)$ ile aralarında asal olacak şekilde e (genel anahtar) sayısı seçilir.
4) $d*e=1 \bmod(\varphi(n))$ olacak şekilde d değeri seçilir. Bu değer özel anahtardır.

5) $c=m^e \ mod(n)$ formülüyle her bir mesaj karakteri şifrelenir.

RSA algoritmasını kullanarak şifrelenen bir mesajı çıkarmak için, ilk 4 adım aynen uygulanır, $m=c^d \ mod(n)$ formülüyle de orijinal mesaj elde edilir.

RSA algoritmasının örnek uygulaması aşağıda verilmiştir:

1) $p=11, q=23$ asal sayıları giriş parametresi olarak seçilmiş olsun.
2) p ve q değerlerini kullanarak, $n=11*23=253$, $\varphi(n)= 10*22=220$ olarak hesaplanır.
3) $e=7$ değeri seçilir (veya 3. maddedeki şartları sağlayan sayılardan bir tanesi seçilir).
4) $d*7=1 \ mod(220)$ eşitliğini sağlayan sayılardan $d=63$ seçilir.
5) Örnek şifrelenecek mesajlar: $187, 47, 74$ değerleri olsun. $c=m^e \ mod(n)$ formülüyle şifrelenmiş mesajlar;
    a. $187^7 \ mod(253) = 209,$
    b. $47^7 \ mod(253) = 185,$
    c. $74^7 \ mod(253) = 178$ olarak hesaplanır.
6) $209, 185, 178$ şifreli mesajlar $m=c^d \ mod(n)$ formülüyle orijinal haline dönüştürülebilir. Orijinal mesajlar;
    a. $209^{63} \ mod(253)= 187,$
    b. $185^{63} \ mod(253)=47$
    c. $178^{63} \ mod(253)=74,$ olarak hesaplanır.

3.5. Kaos Tabanlı Rastgele Sayı Üreteci

Rastgele sayı, rasgele bir şansla makul bir şekilde tahmin edilemeyen, belli bir örüntüde tekrar etmeyen bir dizi sayı veya sembolün üretilmesidir. Rastgele sayı üreteçleri, güvenli haberleşme, veri iletimi ve depolanması gibi birçok alanda kritik öneme sahiptir. Rastgele sayı üreteçleri iki kategoride incelenmektedir: Sözde-rastgele üreteçler ve gerçek rastgele üreteçler.

Sözde rastgele üreteçler, birçok uygulama için yeterli olsa da özellikle bilgi güvenliği alanında yeterli değildir. Çünkü sözde-rastgele üreteçlerin sayıları üretme yöntemleri bilindiği için, üreteceği sayılarda tahmin edilebilmektedir. Bu nedenle çeşitli ataklara karşı dayanıksızdır. Gerçek rastgele üreteçleri, bir sonraki üreteceği sayının tahmin edilemediği veya ürettiği sayıların belli bir örüntü takip etmediği sayı üreteçleridir.

Gerçek rastgele üreteçlerinden biri olan kaos üreteçleri ise giriş parametrelere karşı çok duyarlı oldukları için sürekli zamanda üretecekleri sayılar tahmin

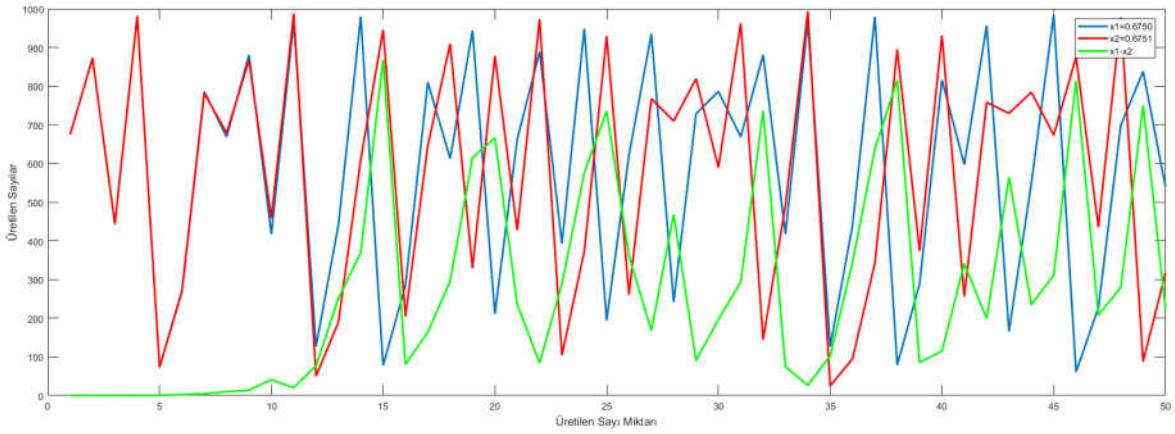
edilememektedir. Bu nedenle özellikle bilgi güvenliği uygulamalarında sıkça kullanılmaktadır. Literatürde çeşitli uygulamalar için üretilmiş farklı birçok kaotik üreteç ya da diğer adıyla kaotik harita vardır. Bunlardan bazıları; Lojistik harita, yuvarlak harita, Bogdanov harita, Baker's harita, Duffing harita, Gauss harita vb.

Literatürde çok kullanılan ve uygulanması basit kaos sistemlerinden bir tanesi lojistik haritadır. Lojistik haritanın formülü şu şekildedir:

$$x_{k+1} = \mu * x_k * (1 - x_k) \tag{3.1}$$

Burada, x_0 ve μ değeri giriş parametreleridir. Lojistik harita, $3.57<\mu\leq4$ olduğu zaman sistem kaotik duruma geçer ve rastgele gibi görülen sayılar üretir.

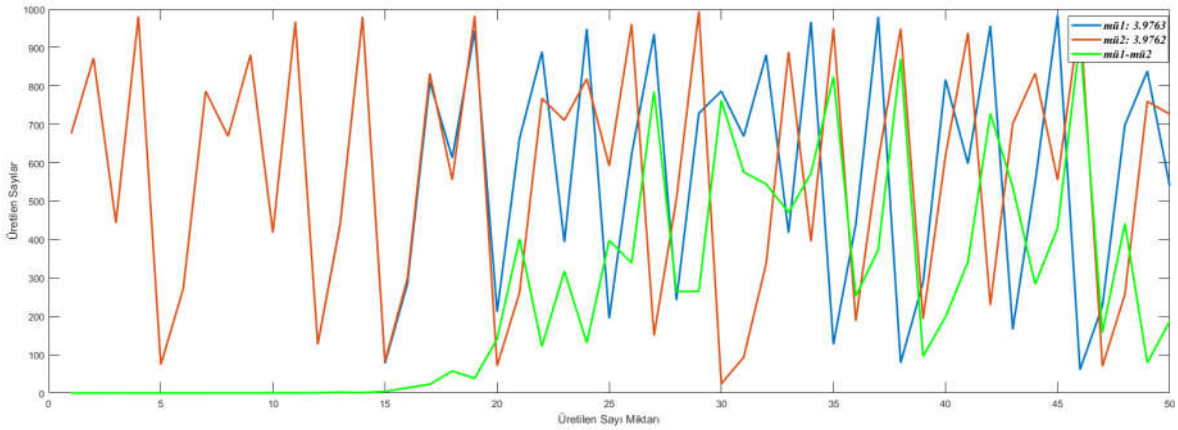
Şekil 3.12'de, lojistik harita ile üretilen sayılara x_0 parametresinin etkisiyle ilgili örnek verilmiştir. Mavi renkli grafikte, $x_0 = 0.6750$ alınarak 50 adet sayı üretilmiştir. Daha sonra x_0 parametresinde çok küçük değişiklik yapılarak $x_0 = 0.6751$ değeri verilmiş ve yine 50 adet sayı üretilmiştir. Kırmızı renkli grafikte bu üretilen sayılar görülmektedir. Her iki durum içinde $\mu = 3.9763$ değeri kullanılmış ve iki durum arasındaki fark grafiği yeşil renkle gösterilmiştir. Görüldüğü gibi üretecin ürettiği ilk 10 sayı birbirine çok yakınken, daha sonraki değerler rastgele gibi görünmektedir.



Şekil 3.12. x_0 parametresinin üretilen sayılara etkisi

Şekil 3.13'de, lojistik harita ile üretilen sayılara μ parametresinin etkisiyle ilgili örnek verilmiştir. Mavi renkli grafikte, $\mu = 3.9763$ alınarak 50 adet sayı üretilmiştir. Daha sonra μ parametresinde çok küçük değişiklik yapılarak $\mu = 3.9762$ değeri verilmiş ve yine 50 adet sayı üretilmiştir. Kırmızı renkli grafikte bu üretilen sayılar görülmektedir. Her iki durum içinde $x_0 = 0.6750$ değeri kullanılmıştır. İki durum arasındaki fark grafiği de yeşil renkle gösterilmiştir. Görüldüğü gibi üretecin ürettiği ilk

15 sayı birbirine yakınken, daha sonraki değerlerin birbirinden çok farklı olduğu ve belirli bir sırayı izlemediği çıkarımı yapılabilir.



Şekil 3.13. μ parametresinin üretilen sayılara etkisi

3.6. Görüntü Değerlendirme Kriterleri

Görüntü değerlendirme kriterleri, bir görüntüde değişim olup olmadığını veya değişimin miktarını öğrenmek için kullanılan yöntemlerdir. Bu tezde kullanılan görüntü değerlendirme kriterleri aşağıda açıklanmıştır.

3.6.1. Ortalama karesel hata

Görüntü kalitesi hakkında bilgi veren, en bilinen yöntemlerden bir tanesi ortalama karesel hata (MSE - Mean Square Error)'dır. MSE, orijinal görüntü ile stego görüntü arasındaki farkların karelerinin toplamının ortalamasına eşittir. Bu nedenle, düşük MSE değeri görüntü de az bozulma olduğu, yüksek MSE değeri ise görüntü de daha fazla bozulma olduğu anlamına gelmektedir.

MSE formülü şu şekildedir:

$$MSE = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{[S(i,j) - I(i,j)]^2}{m * n} \tag{3.2}$$

Burada *m,n* görüntü çözünürlüğünü, *S(i,j)* stego görüntü piksellerini, *I(i,j)* ise orijinal görüntü piksellerini ifade etmektedir.

3.6.2. PSNR

PSNR (Peak Signal to Noise Ratio – Tepe Sinyal Gürültü Oranı) görüntü kalitesini değerlendirmede kullanılan önemli kriterlerden bir tanesidir. PSNR, orijinal görüntünün mümkün olan en yüksek gücünün orijinal görüntü ve stego görüntüsü arasındaki farkların gücüne oranıdır. Yüksek PSNR değeri görüntüde az bozulma

olduğu, düşük PSNR değeri ise görüntüde daha fazla bozulma olduğu anlamına gelmektedir.

PSNR değeri şu formülle hesaplanabilir:

$$PSNR = 20 * \log_{10} \frac{255}{\sqrt{MSE}} \tag{3.3}$$

3.6.3. Ortalama fark

Ortalama fark (AD-Average Difference), orijinal görüntü pikselleri ile stego görüntü pikselleri arasındaki farkların toplamının ortalamasıdır. Düşük ortalama fark değeri, görüntüde bozulmanın daha az olduğu anlamına gelmektedir.

Ortalama fark formülü şu şekildedir:

$$AD = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{[I(i,j) - S(i,j)]}{m * n} \tag{3.4}$$

3.6.4. Normalize mutlak hata

Normalize Mutlak Hata (NAE- Normalized Absolute Error), görüntü kalitesini belirlemek için kullanılan diğer bir yöntemdir. NAE, orijinal görüntü ve stego görüntüsü arasındaki mutlak farkın orijinal görüntüye oranıdır. NAE değeri 0'a ne kadar yakınsa, iki görüntü birbirine o kadar çok benziyor anlamına gelmektedir.

NAE'nin formülü aşağıda verilmiştir:

$$NAE = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} |I(i,j) - S(i,j)|}{\sum_{i=1}^{m} \sum_{j=1}^{n} I(i,j)} \tag{3.5}$$

3.6.5. Normalize çapraz korelasyon

Normalize Çapraz Korelasyon (NCC – Normalized Cross Correlation), bir görüntünün diğerine göreceli benzerlik oranıdır. NCC değeri 1'e yakınsa, bu iki görüntünün birbiriyle daha çok ilişkili olduğu anlamına gelir.

NCC formülü aşağıda verilmiştir:

$$NCC = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} I(i,j) * S(i,j)}{\sum_{i=1}^{m} \sum_{j=1}^{n} I(i,j) * I(i,j)} \tag{3.6}$$

3.6.6. Yapısal benzerlik testi

Yapısal benzerlik testi (SSIM, Structural Similarity Index Measure), iki görüntünün birbirine yapısal olarak ne kadar benzediğini gösteren önemli bir

değerlendirme kriteridir. 0 ile 1 arasında değer almaktadır. 1, iki resmin tamamen aynı olduğunu, 0 ise iki resmin tamamen farklı olduğunu göstermektedir.

SSIM değeri şu formülle hesaplanabilir:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.7}$$

Burada, *x, y* karşılaştırılan iki resmi, $\{\mu_{x,y}, \sigma_{x,y}, C_{1,2}\}$ sırasıyla piksel yoğunluk ortalaması, standart sapma ve katsayıları ifade etmektedir.

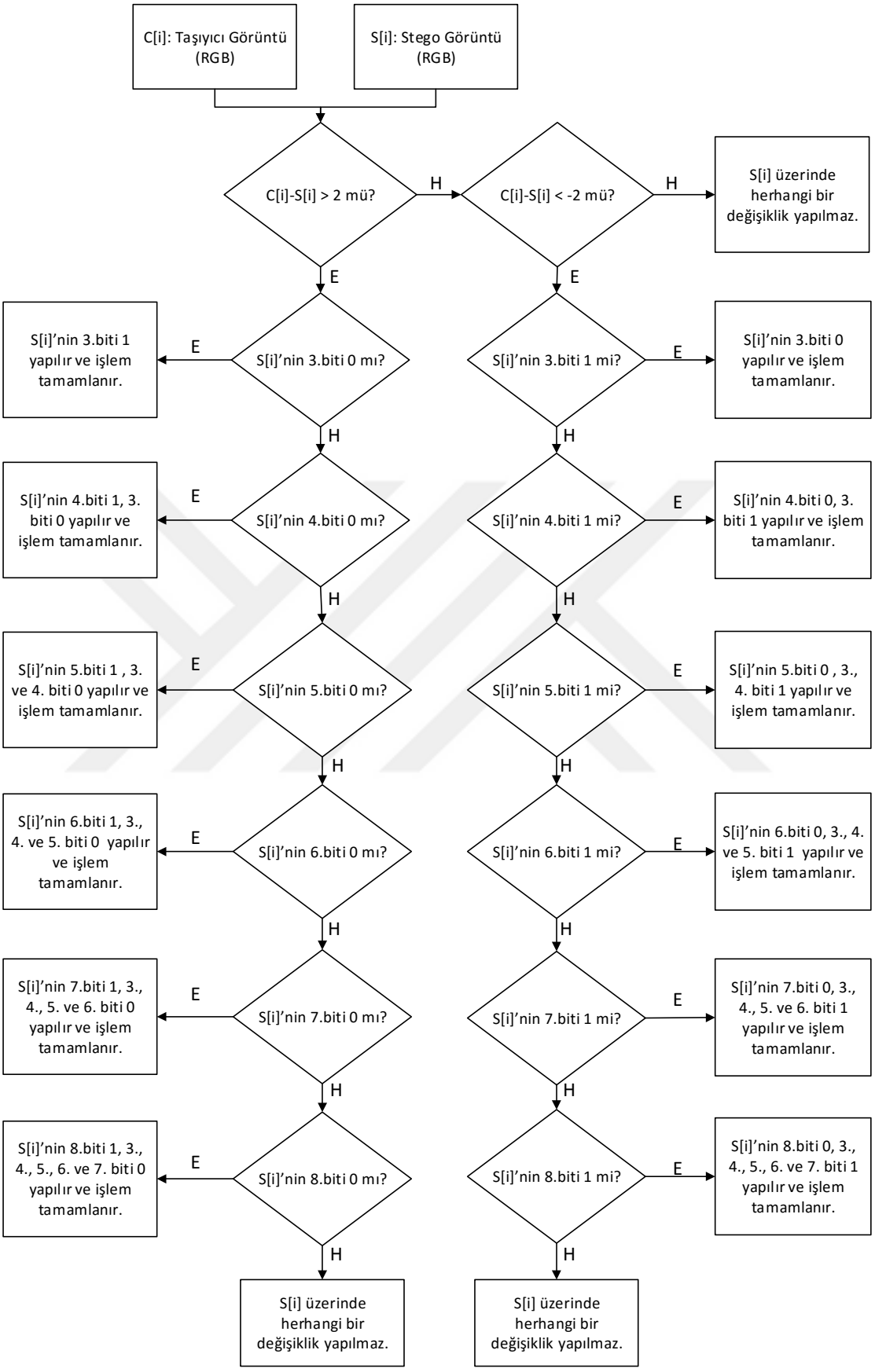
4. ÖNERİLEN YÖNTEM

1-LSB, 2-LSB ve 3-LSB yöntemleri, uygulanmasının kolay olması, yüksek kapasiteli olması nedeniyle steganografi'de kullanılan en çok yöntemlerdendir. Geliştirilen birçok farklı yöntem de bu yöntemler üzerine inşa edilmektedir. Fakat tespit edilmesinin kolay olması, tespit edildiği zaman 3. kişilerin mesaja doğrudan ulaşabilmesi nedeniyle geliştirilmeye ihtiyaçları bulunmaktadır.

Bu bölümde, 2-LSB ve 3-LSB yöntemlerini modifiye ederek oluşturduğumuz, mevcut sıkıştırma ve şifreleme algoritmalarıyla geliştirdiğimiz yeni yöntemler hakkında ayrıntılı bilgi verilmiştir. Önerilen yöntemlerin temeli, 2-LSB ve 3-LSB değiştirme yöntemlerini uygularken taşıyıcı ve stego görüntü pikselleri arasında oluşan fazla değişimi azaltmaya dayanmaktadır. Önerdiğimiz yöntem, 1-LSB yöntemi üzerinde etkisi olmaması, 4 ve daha fazla bit değiştirme yöntemlerinin literatürde kullanımları olmaması nedeniyle sadece 2-LSB ve 3-LSB yöntemlerine uygulanmıştır. Önerdiğimiz 2-LSB yönteminin uygulama adımları Şekil 4.1'de, 3-LSB yönteminin uygulama adımları Şekil 4.2'de verilmiştir.

4.1. Önerilen 2-LSB Gömme Yöntemi

Şekil 4.1'de de görülebileceği gibi mesaj, taşıyıcı görüntüye klasik 2-LSB yöntemiyle gömülür. Gömme işlemi sonunda, taşıyıcı görüntü ile stego görüntü pikselleri arasındaki fark incelenir. Her piksele 2 bit gömüldüğü için, fark 3, 2, 1, 0, -1, -2, -3 değerlerinden birini alacaktır. Eğer pikseller arasındaki fark -2, -1, 0, 1 veya 2 ise, o piksele herhangi bir işlem yapılmaz, aynen bırakılır. Eğer fark 3 ise, o pikselin 3. bitinden itibaren bütün bitleri incelenir, karşılaşılan ilk 0 rakamı 1'e dönüştürülür, daha önceki 1 rakamları ise 0'a dönüştürülür. Böylece o piksele ondalık 4 sayısı eklenmiş olur ve pikseller arasındaki fark 3'den -1'e düşer. Eğer 3. ile 8. bitleri arasında hiç 0 rakamı yok ise o piksel aynen bırakılır ve değişiklik yapılmaz. Eğer fark -3 ise, o pikselin 3. bitinden itibaren bütün bitleri incelenir, karşılaşılan ilk 1 rakamı 0'a dönüştürülür, daha önceki 0 rakamları ise 1'e dönüştürülür. Böylece o pikselden ondalık 4 sayısı çıkarılmış olur ve pikseller arasındaki fark -3'den 1'e düşer. Eğer 3. ile 8. bitleri arasında hiç 1 rakamı yok ise o piksel aynen bırakılır ve değişiklik yapılmaz. Bu yöntem ile bozulmanın fazla olduğu piksellerde ki bozulma azaltılmış olur. Çizelge 4.1'de önerilen 2-LSB'nin uygulanmasıyla ilgili örnekler verilmiştir.



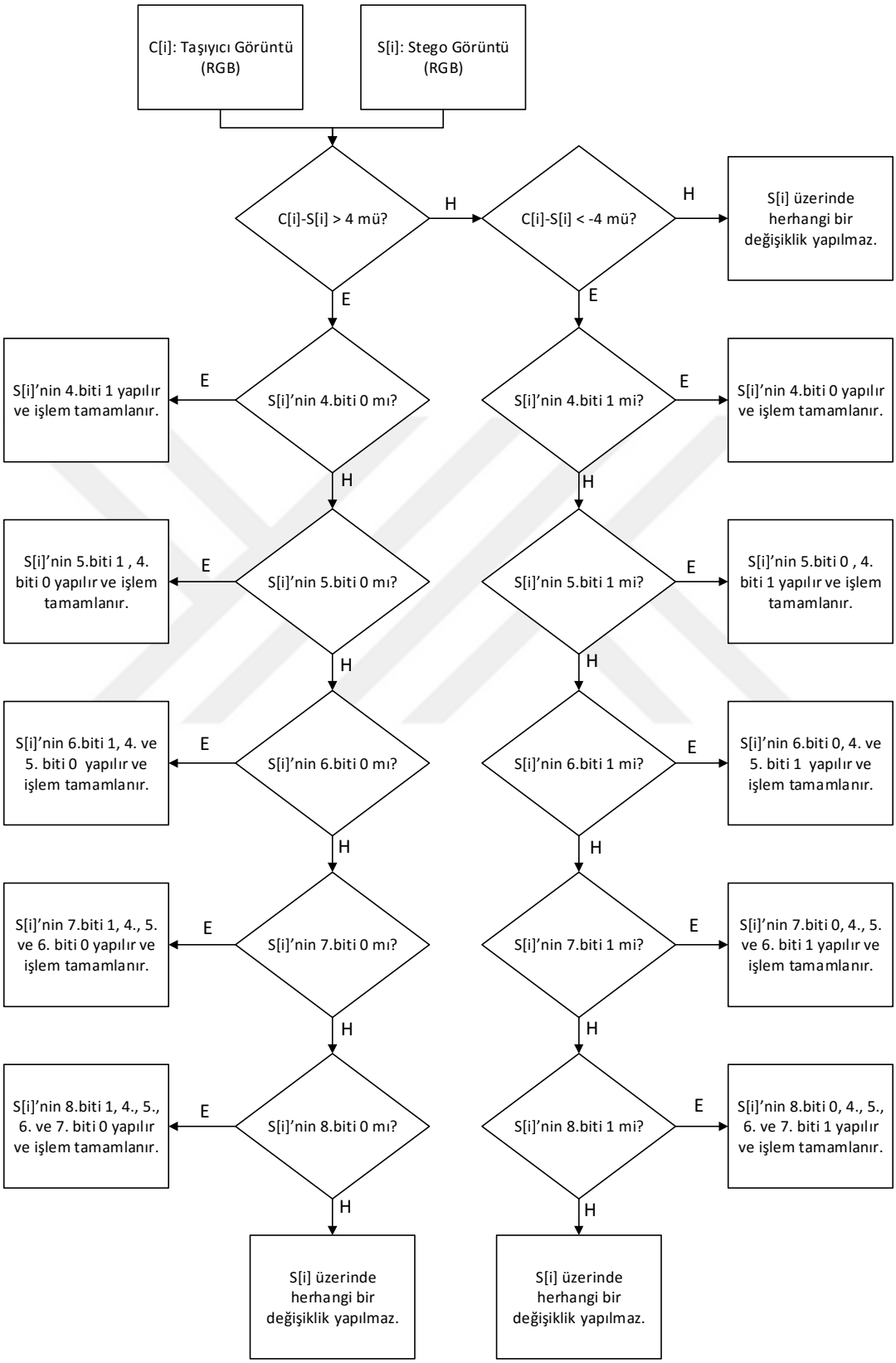
Şekil 4.1. Önerilen 2-LSB yönteminin akış diyagramı

Çizelge 4.1. Önerilen 2-LSB yönteminin örnek uygulaması

Taşıyıcı Görüntü	Gömülecek Mesaj (2-bit)	Stego Görüntü	Taşıyıcı-Stego Farkı	Yeni Stego Görüntü	Taşıyıcı-Yeni Stego Farkı
01111001	01	011110***01***	0	01111001	0
00100011	10	001000***10***	1	00100010	1
10010000	11	100100***11***	**-3**	100***01***11	***1***
00000001	10	000000***10***	-1	00000010	-1
11000011	00	110000***00***	**3**	11000***1***00	***-1***
11000010	11	110000***11***	-1	11000011	-1
00101110	01	001011***01***	1	00101101	1
11111000	01	111110***01***	-1	11111001	-1
00110000	10	001100***10***	-2	00110010	-2
11010100	11	110101***11***	**-3**	11010***0***11	***1***
11100000	11	111000***11***	**-3**	11***011***11	***1***
00100000	10	001000***10***	-2	00100010	-2
01010001	01	010100***01***	0	01010001	0
01101101	10	011011***10***	-1	01101110	-1
11101001	10	111010***10***	-1	11101010	-1
11001101	01	110011***01***	0	11001101	0
01110111	00	011101***00***	**3**	0111***1***00	***-1***
01111110	10	011111***10***	0	01111110	0
01001101	10	010011***10***	-1	01001110	-1
10000011	10	100000***10***	1	10000010	1

Çizelge 4.1'de 8'er bit ve 20 pikselden oluşan rastgele oluşturulmuş taşıyıcı görüntü üzerine, yine rastgele oluşturulmuş 2'şer bitlik mesajlar 2-LSB yöntemiyle gizlenmiştir. Gizleme işleminden sonra taşıyıcı görüntü ile stego görüntü pikselleri arasındaki farklara bakılmış, fark -3 veya 3 ise önerilen yöntem uygulanmıştır. Tabloda da görüleceği gibi, 20 pikselin 5 tanesinde fark -3 veya 3 olmuş, bu nedenle bu 5 piksel dışındaki piksellerde değişiklik yapılmamıştır. Yapılan iyileştirme işleminden önce ortalama piksel farkı 1.2 iken, iyileştirme işleminden sonra bu fark 0.9'e düşürülmüş, böylece görüntüdeki bozulma miktarının azaltılması sağlanmıştır.

4.2. Önerilen 3-LSB Gömme Yöntemi



Şekil 4.2. Önerilen 3-LSB yönteminin akış diyagramı

Şekil 4.2'de de görülebileceği gibi mesaj, taşıyıcı görüntüye klasik 3-LSB yöntemiyle gömülür. Gömme işlemi sonunda, taşıyıcı görüntü ile stego görüntü pikselleri arasındaki fark incelenir. Her piksele 3 bit gömüldüğü için, fark -7 ile 7 arasındaki değerlerden birini alacaktır. Eğer pikseller arasındaki fark -4, -3, -2, -1, 0, 1, 2, 3 veya 4 ise, o piksele herhangi bir işlem yapılmaz, aynen bırakılır. Eğer fark 5, 6 veya 7 ise, o pikselin 4. bitinden itibaren bütün bitleri incelenir, karşılaşılan ilk 0 rakamı 1'e dönüştürülür, daha önceki 1 rakamları ise 0'a dönüştürülür. Böylece o piksele ondalık 8 sayısı eklenmiş olur ve pikseller arasındaki fark sırasıyla 5, 6, 7'den -3, -2 ve -1'e düşer. Eğer 4. ile 8. bitleri arasında hiç 0 rakamı yok ise o piksel aynen bırakılır ve değişiklik yapılmaz. Eğer fark -5, -6 veya -7 ise, o pikselin 4. bitinden itibaren bütün bitleri incelenir, karşılaşılan ilk 1 rakamı 0'a dönüştürülür, daha önceki 0 rakamları ise 1'e dönüştürülür. Böylece o pikselden ondalık 8 sayısı çıkarılmış olur ve pikseller arasındaki fark sırasıyla -5, -6, -7'den 3, 2 ve 1'e düşer.

Çizelge 4.2. Önerilen 3-LSB yönteminin örnek uygulaması

Taşıyıcı Görüntü	Gömülecek Mesaj (3-bit)	Stego Görüntü	Taşıyıcı-Stego Farkı	Yeni Stego Görüntü	Taşıyıcı-Yeni Stego Farkı
01111001	111	01111***111***	**-6**	0111***0***111	***2***
00100011	001	00100***001***	2	00100001	2
10010000	111	10010***111***	**-7**	100***01***111	1
00011111	010	00011***010***	**5**	00***100***010	***-3***
11000011	100	11000***100***	-1	11000100	-1
11000010	111	11000***111***	**-5**	1***0111***111	***3***
00101110	111	00101***111***	-1	00101111	-1
11111000	001	11111***001***	-1	11111001	-1
00110000	010	00110***010***	-2	00110010	-2
11010100	110	11010***110***	-2	11010110	-2
11100111	000	11100***000***	7	1110***1***000	-1
00100000	101	00100***101***	**-5**	00***011***101	***3***
01010001	101	01010***101***	-4	01010101	-4
01101101	101	01101***101***	0	01101101	0
11101001	110	11101***110***	**-5**	1110***0***110	***3***
11001101	100	11001***100***	1	11001100	1
01110111	001	01110***001***	**6**	0111***1***001	***-2***
01111110	111	01111***111***	-1	01111111	-1
01001111	001	01001***001***	**6**	010***10***001	***-2***
10000011	001	10000***001***	2	10000001	2

Eğer 4. ile 8. bitleri arasında hiç 1 rakamı yok ise o piksel aynen bırakılır ve değişiklik yapılmaz. Bu yöntem ile bozulmanın fazla olduğu piksellerde ki bozulma azaltılmış olur. Çizelge 4.2'de önerilen 3-LSB'nin uygulanmasıyla ilgili örnekler verilmiştir.

Çizelge 4.2'de 8'er bit ve 20 pikselden oluşan rastgele oluşturulmuş taşıyıcı görüntü üzerine, yine rastgele oluşturulmuş 3'şer bitlik mesajlar 3-LSB yöntemiyle gizlenmiştir. Gizleme işleminden sonra taşıyıcı görüntü ile stego görüntü pikselleri arasındaki farklara bakılmış, fark -7, -6, -5, 5, 6 veya 7 ise önerilen yöntem uygulanmıştır. Tabloda da görüleceği gibi, 20 pikselin 9 tanesinde fark -7, -6, -5, 5, 6 veya 7 olmuş, bu nedenle bu 9 piksel dışındaki piksellerde değişiklik yapılmamıştır. Yapılan iyileştirme işleminden önce ortalama piksel farkı 3.45 iken, iyileştirme işleminden sonra bu fark 1.85'e düşürülmüş, böylece görüntüdeki bozulma miktarının azaltılması sağlanmıştır.

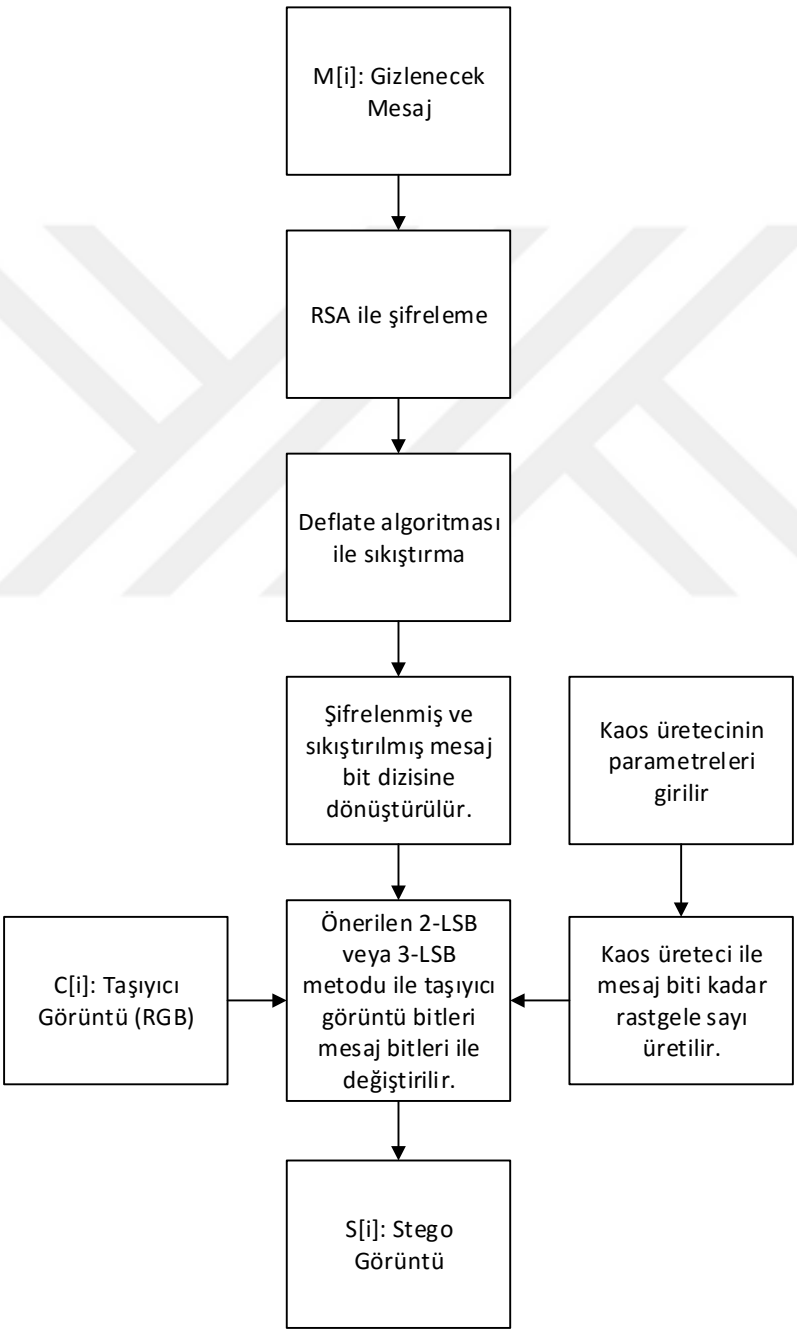
Önerilen yöntem ile piksellerdeki bozulma azaltılmış, bu sayede steganografi'nin 3 temel ilkesinden biri olan algılanamazlık ilkesinde iyileşme sağlanmıştır. Fakat steganografi'nin diğer ilkeleri olan güvenlik ve kapasitede herhangi bir değişim olmamıştır. Bu iki kriterde de iyileştirme sağlayabilmek adına, mevcut sıkıştırma ve şifreleme yöntemleri, önerilen yöntemle birleştirilmiştir.

Önerilen 2-LSB veya 3-LSB yönteminde gömme işlemi sırayla yapıldığı için, iletişimi izleyen 3. kişi şüphelendiği takdirde mesajı çıkarmaya çalışırsa kolaylıkla çıkarabilecektir. Bu sorunu çözebilmek ve önerdiğimiz yöntemi geliştirmek adına iki farklı yol birlikte uygulanmıştır: Mesajı gömmeden önce şifrelemek ve mesajı sıralı değil rastgele gömmek.

Mesajı taşıyıcı resme gömmeden önce şifrelemek, 3. kişiler görüntü içinden gömülen bilgiyi tam olarak çıkarabilse bile, orijinal mesaja ulaşmayı zorlaştıracaktır. Bu amaçla mesaj RSA şifreleme yöntemi ile şifrelenmiştir. RSA algoritmasının seçilmesindeki neden, yaygın kullanılması ve asimetrik bir şifreleme algoritması olmasıdır. RSA şifrelemede genel ve özel anahtar üretmek için gerekli olan (p, q) değerleri giriş parametresi olarak kullanılmıştır. Bu nedenle alıcı tarafın orijinal mesaja ulaşabilmesi için p ve q değerlerinin alıcı tarafa iletilmesi gerekmektedir.

Mesajı taşıyıcı görüntü üzerine gömerken sıralı gömmek, mesajın 3. kişilerce çıkarılabilmesini kolaylaştırmaktadır. Mesajı gömmeden önce şifrelemek bu duruma büyük oranda çözüm olsa da, gömme işlemi sıralı olduğu için, 3. kişiler şifreli mesajı rahatlıkla elde edebilecektir. Kriptoloji yöntemleri, kırılması zor yöntemler olsa da kırmak imkânsız değildir. Bu nedenle orijinal mesajın başka kişilerin eline geçme

ihtimali bulunmaktadır. Bu sorunu aşmak için steganografi'de mesaj genellikle sıralı olarak değil, çeşitli rastgele sayı üreteçleriyle görüntü içine rastgele gömülmektedir. Önerdiğimiz yöntemde, bu sıkıntıyı aşmak için lojistik harita tabanlı kaos üreteci ile rastgele sayılar üretilmiş ve mesaj görüntü içerisine rastgele gömülmüştür. Kaos üretecinin x_0 ve μ değerleri giriş parametresi olarak kullanılmıştır. Bu nedenle gizlenen mesajın çıkarılabilmesi için bu parametrelerin alıcı tarafa iletilmesi gerekmektedir.



Şekil 4.3. Önerilen yöntemi genel akış diyagramı

Mesajı taşıyıcı görüntü gömmeden önce sıkıştırmak, gömülecek mesaj biti sayısını azaltacağı için stego görüntüdeki bozulmayı azaltacak, aynı zamanda taşıyıcı üzerine gömülebilecek mesaj uzunluğunu, yani görüntünün kapasitesini de artıracaktır. Ayrıca kullanılan sıkıştırma algoritması, 3. kişiler tarafından bilinmeyeceği için, stego görüntüye gizlenen veri okunsa bile, orijinal mesaja doğrudan ulaşılamayacaktır. Bu amaçla, mesaj gömülmeden önce, metin sıkıştırma algoritmaları kullanarak sıkıştırılmış, daha sonra gömülmüştür. Metin sıkıştırma algoritması olarak da LZW, Arithmetic, MTF ve Deflate olmak üzere 4 farklı algoritma denenmiş, kendi aralarında karşılaştırma yapılmıştır.

Önerilen yöntemin genel akış diyagramı Şekil 4.3'de verilmiştir.

4.3. Arayüz

Mevcut ve önerilen algoritmaların uygulaması için MATLAB üzerinde yazılan kodlar, kullanım kolaylığı açısından yine MATLAB GUI'de arayüz haline getirilmiştir. Tasarlanan arayüz mesaj gizleme ve mesaj çıkarma olarak iki ayrı bölümden oluşmaktadır.

4.3.1. Mesaj gizleme arayüzü

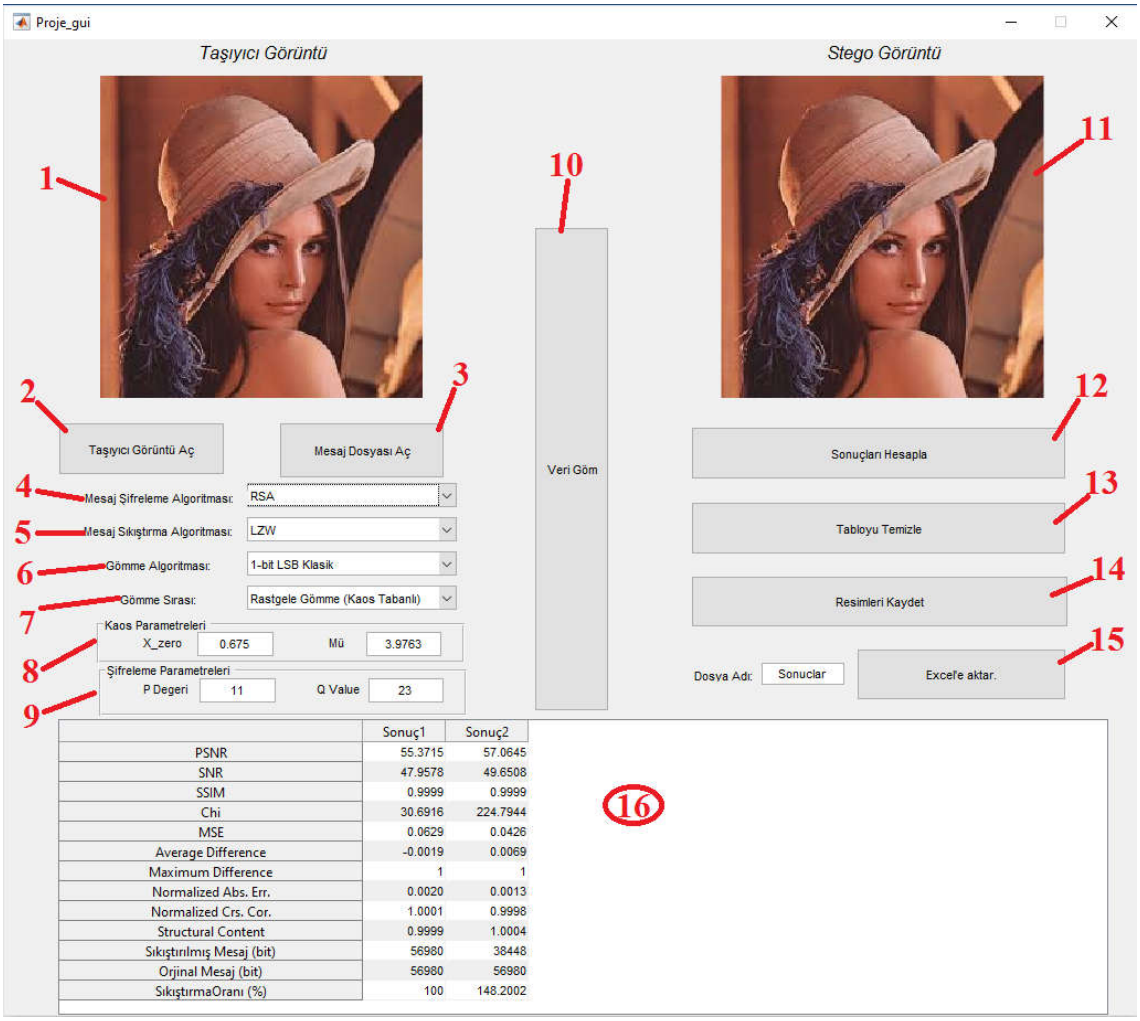
Mesaj gizleme arayüzü, mevcut algoritmalar ve bir önceki bölümde bahsedilen önerilen yöntemlerin uygulandığı arayüzdür. Şekil 4.4'de bu arayüz görülmektedir.

1 numara ile gösterilen bölüm, mesajın gömüleceği taşıyıcı görüntünün arayüz üzerinde gösterildiği bölümdür.

2 numara ile gösterilen buton ile herhangi bir klasörde bulunan taşıyıcı görüntü seçilir ve 1 numaralı bölümde gösterilmektedir. Butona tıklandığı zaman sadece uygun dosya türündeki görüntüler görülebilmektedir.

3 numara ile gösterilen buton ile herhangi bir klasörde bulunan taşıyıcı resme gömülecek metin dosyası seçilmektedir. Butona tıklandığı zaman sadece ".txt" uzantılı metin dosyaları görülmektedir.

4 numaralı bölümde, mesajı seçildikten sonra şifrelenip şifrelenmeyeceği seçilmektedir. "Şifreleme yok" ve "RSA" olmak üzere iki seçenek bulunmaktadır. "Şifreleme Yok" seçilirse mesaj şifrelenmeden direk gömme veya sıkıştırma işlemine tabi tutulmaktadır. "RSA" seçildiğinde ise 9 ile gösterilen bölümde RSA giriş parametreleri olan p ve q değerlerinin girilebileceği bir bölüm açılmakta ve bu değerlere göre mesaj şifrelenmektedir.



Şekil 4.4. Mesaj gizleme arayüzü

5 numaralı bölümde, 4 numaralı bölümdeki seçime göre, şifrelenen veya şifrelenmeyen mesaj, bu bölümde seçilen algoritmayla sıkıştırılmaktadır. Bu bölümdeki seçenekler, "Sıkıştırma Yok", "LZW", "Arithmetic", "MTF", "Deflate" olmak üzere 5 tanedir. Eğer "Sıkıştırma Yok" seçilirse, mesaj sıkıştırılmadan aynen gömülmekte, diğer durumlarda seçilen algoritmayla sıkıştırılmaktadır.

6 numaralı bölümde, gömme işleminin hangi algoritmayla yapılacağı seçilmektedir. Bu bölümde, "1-bit Klasik LSB", "2-bit Klasik LSB", "3-bit Klasik LSB", "2-bit Önerilen LSB", "3-bit Önerilen LSB" seçenekleri mevcut olup, gömme işlemi seçilen algoritmayla yapılmaktadır.

7 numaralı bölümde, mesajın taşıyıcı görüntü üzerine hangi sırayla gömüleceği seçilmektedir. Bu bölümdeki seçenekler, "Sıralı Gömme" ve "Rastgele Gömme"dir. "Sıralı Gömme" seçildiği zaman, mesaj bitleri taşıyıcı piksellerin ilkinden başlayarak sırayla gömülmektedir. "Rastgele Gömme" seçildiğinde ise mesaj bitleri taşıyıcı

görüntü üzerine lojistik harita tabanlı kaos üretecinde üretilen sayılara göre rastgele gömülmektedir. Bu seçenek seçildiğinde, lojistik haritanın başlangıç parametreleri olan x_0 ve μ değerlerinin girilebileceği, 8 numara ile gösterilen bölüm açılmaktadır.

8 numaralı bölüm, lojistik haritanın başlangıç parametrelerinin girileceği bölümdür. 7. bölümde "Rastgele Gömme" seçildiği zaman aktif olmaktadır.

9 numaralı bölüm, RSA şifreleme algoritmasının giriş parametrelerinin girildiği bölümdür. 4. bölümde "RSA" seçildiği takdirde aktif olmaktadır.

10 numara ile gösterilen buton, veri gömme işlemini başlatan butondur. İlk 9 bölümde seçilen ayarlara göre mesaj taşıyıcı görüntü üzerine gömülmektedir. Gömme işlemi tamamlandığında, 11 numaralı bölümde stego görüntü otomatik olarak gösterilmektedir.

12 numaralı buton, gömme işlemi tamamlandıktan sonra aktif olmakta ve o anki taşıyıcı ve stego görüntüyü görüntü kalitesi değerlendirme kriterlerine göre karşılaştırıp sonuçları 16 numara ile gösterilen tabloya yazmaktadır.

13 numaralı buton tablodaki tüm verileri temizlemek için kullanılmaktadır.

14 numaralı buton o anki taşıyıcı ve stego görüntüyü otomatik isimlendirerek kaydetmektedir.

15 numaralı buton o anki tablodaki sonuçları belirtilen isimde excel dosyasına kaydetmektedir.

16 numaralı bölüm, 12 numaralı butona basıldığında hesaplanan sonuçların gösterildiği bölümdür. Butona basıldıkça yeni sütun oluşturularak sonuçları göstermeye devam etmektedir. Sütun sayısı sınırlı değildir. Tablodaki veriler 13 numaralı buton ile silinebilmektedir.

4.3.2. Mesaj çıkarma arayüzü

Mesaj çıkarma arayüzü, mevcut algoritmalar ve bir önceki bölümde bahsedilen önerilen yöntemlerle oluşturulan stego görüntülerin çözüldüğü ve gizlenen mesajın çıkarıldığı arayüzdür. Şekil 4.5'de bu arayüz görülmektedir.

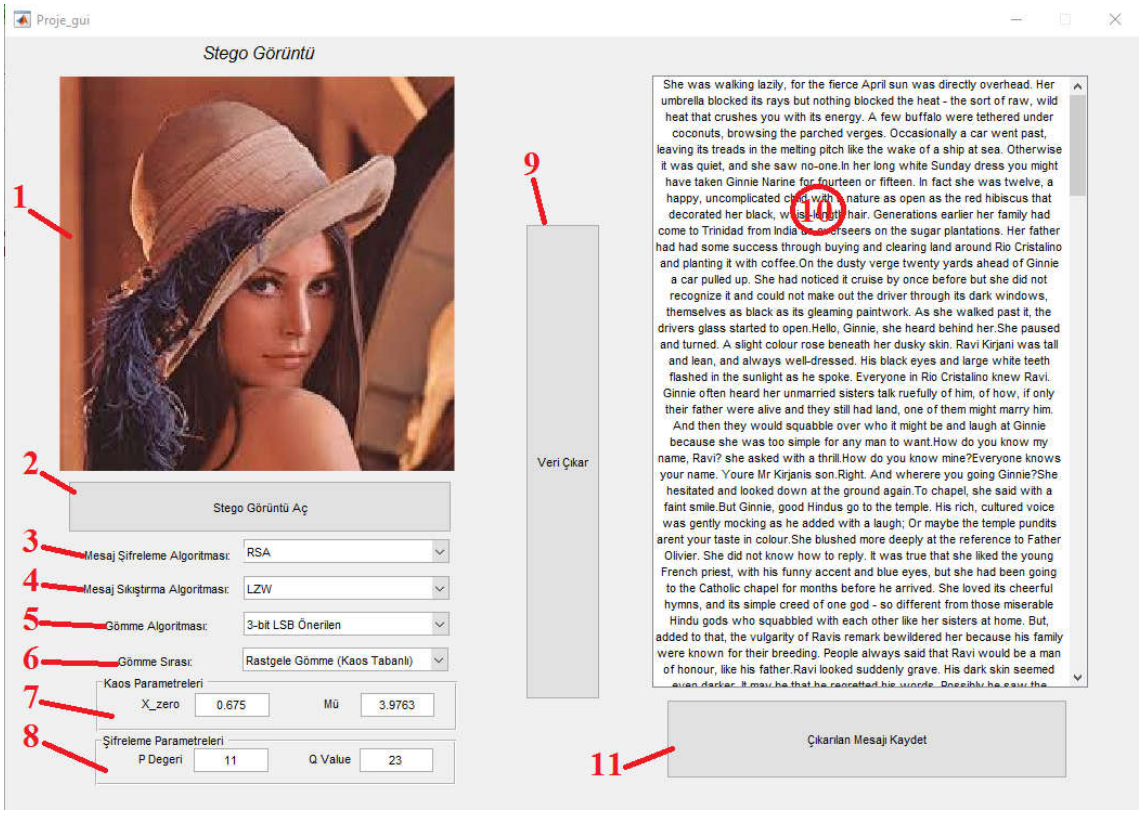
1 numara ile gösterilen bölüm, içerisinde mesajın gömülü olduğu stego görüntünün arayüz üzerinde gösterildiği bölümdür.

2 numara ile gösterilen buton ile herhangi bir klasörde bulunan stego görüntü seçilip ve 1 numaralı bölümde gösterilmektedir. Butona tıklandığı zaman sadece uygun dosya türündeki görüntüler görülebilmektedir.

3 numaralı bölümde, stego görüntü içindeki mesajın gömülmeden önce şifrelenip şifrelenmediği belirlenir. "Şifreleme yok" ve "RSA" olmak üzere iki seçenek

bulunmaktadır. “Şifreleme Yok” seçilirse mesaj şifre çözme işleminden geçmeden direk ekranda gösterilir. “RSA” seçildiğinde ise 9 ile gösterilen bölümde RSA giriş parametreleri olan *p* ve *q* değerlerinin girilebileceği bir bölüm açılmakta ve bu değerlere göre şifreli mesaj çözülmektedir.

4 numaralı bölümde, mesajın taşıyıcı görüntüye gömülmeden önce sıkıştırılıp sıkıştırılmadığı belirlenmektedir. Bu bölümdeki seçenekler, “Sıkıştırma Yok”, “LZW”, “Arithmetic”, “MTF”, “Deflate” olmak üzere 5 tanedir. Eğer “Sıkıştırma Yok” seçilirse, mesaj sıkıştırılmadan gömülmüş, diğer durumlarda seçilen algoritmayla sıkıştırılmış anlamına gelmektedir.



Şekil 4.5. Mesaj çıkarma arayüzü

5 numaralı bölümde, mesaj çıkarma işleminin hangi algoritmayla yapılacağı seçilmektedir. Bu bölümde, “1-bit Klasik LSB”, “2-bit Klasik LSB”, “3-bit Klasik LSB”, “2-bit Önerilen LSB”, “3-bit Önerilen LSB” seçenekleri mevcut olup, çıkarma işlemi seçilen algoritmayla yapılmaktadır.

6 numaralı bölümde, mesajın taşıyıcı görüntü üzerine hangi sırayla gömüldüğü seçilmektedir. Bu bölümdeki seçenekler, “Sıralı Gömme” ve “Rastgele Gömme”dir. “Sıralı Gömme” seçildiği zaman, mesaj bitleri taşıyıcı piksellerin ilkinden başlayarak

sırayla okunmaktadır. “Rastgele Gömme” seçildiğinde ise mesaj bitleri taşıyıcı görüntü üzerine lojistik harita tabanlı kaos üretecinde üretilen sayılara göre okunmaktadır. Bu seçenek seçildiğinde, lojistik haritanın başlangıç parametreleri olan x_0 ve μ değerlerinin girilebileceği, 7 numara ile gösterilen bölüm açılmaktadır.

7 numaralı bölüm, lojistik haritanın başlangıç parametrelerinin girileceği bölümdür. 6. bölümde “Rastgele Gömme” seçildiği zaman aktif olmaktadır.

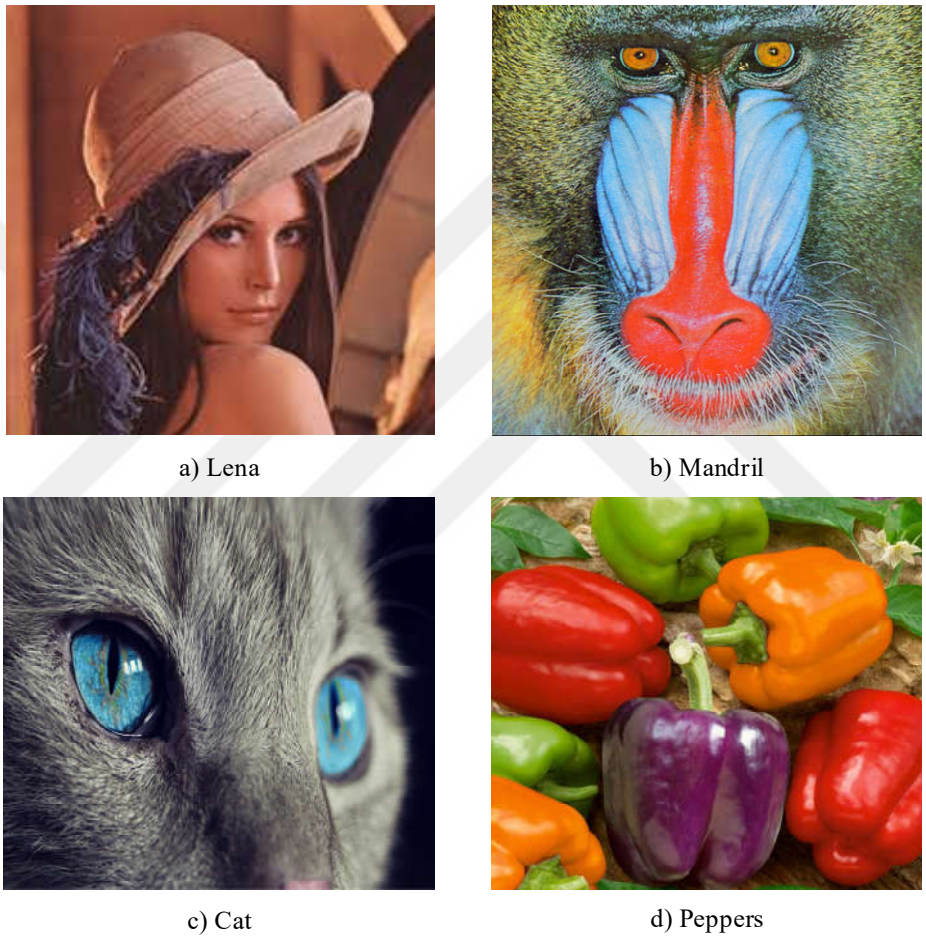
8 numaralı bölüm, RSA şifreleme algoritmasının giriş parametrelerinin girildiği bölümdür. 3. bölümde “RSA” seçildiği takdirde aktif olmaktadır.

9 numara ile gösterilen buton, veri çıkarma işlemini başlatan butondur. İlk 8 bölümde seçilen ayarlara göre mesaj stego görüntüden çıkarılmaya çalışılır. Çıkarma işlemi tamamlandığında, 10 numaralı bölümde çıkarılan mesaj otomatik olarak gösterilmektedir. Eğer çıkarma ayarları doğru yapılmasa, orijinal mesaj görüntülenemeyecektir.

11 numaralı buton, çıkarılan mesajı “.txt” uzantılı dosyaya kaydetmektedir.

5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu bölümde mevcut ve önerilen algoritmanın uygulanması sonucu elde edilen sonuçlar değerlendirilmiştir. Mevcut ve önerilen algoritmaları test etmek için 3 farklı metin dosyası, 4 farklı görüntü kullanılmıştır. Kullanılan taşıyıcı görüntüler Şekil 5.1'de verilmiştir. Bu görüntülerden, "Lena" 225x225, "Mandrill" 512x512, "Cat" 960x603, "Peppers" 600x600 piksel çözünürlüğe sahip RGB 24-bit görüntülerdir. Bu görüntüler ".bmp" dosya formatındadır.



a) Lena b) Mandril

c) Cat d) Peppers

Şekil 5.1. Taşıyıcı görüntüler

Görüntülere gömülecek mesaj olarak farklı uzunluklarda 3 metin dosyası kullanılmıştır. Bu metin dosyalarının uzunlukları sırasıyla, 6.95 kB, 13.59 kB ve 17.13 kB büyüklüğündedir.

5.1. Metin Sıkıştırma Yöntemlerinin Karşılaştırılması

Önerilen yöntemde mesaj dosyası, görüntü üzerine gizlenmeden önce farklı sıkıştırma yöntemleri kullanılarak sıkıştırılmış, sıkıştırma sonucu oluşan yeni mesaj

uzunluğu ve sıkıştırma oranları Çizelge 5.1'de verilmiştir. Bu tezde hesaplanan sıkıştırma oranı değerleri şu formüle göre hesaplanmıştır:

$$\%\ Sıkıştırma = \frac{Giren\ Dosya\ Boyutu}{Çıkan\ Dosya\ Boyutu}$$

Çizelge 5.1. Sıkıştırma oranları

Dosya Adı	Sıkıştırılmamış Uzunluk (bit)	LZW	Arithmetic	MTF	Deflate
		Sıkıştırma Oranı			
Text-1 dosyası	56980	38448 % 148.20	35448 %160.74	48840 %116.67	29496 %193.18
Text-2 dosyası	111405	72969 %152.67	69361 %160.62	95490 %116.67	56832 %196.03
Text-3 dosyası	140364	84643 %165.83	88109 %159.31	120312 %116.67	65744 %213.50

Elde edilen sonuçları incelediğimiz zaman, 56980, 111405 ve 140364 bit uzunluğunda olan orijinal mesaj dosyaları Lempel-Ziv-Welch algoritmasıyla sıkıştırıldığında sırasıyla, 38448, 72969 ve 84643 bit boyutuna düşmüştür. Sıkıştırma oranları ise sırasıyla %148.2, %152.67 ve % 165.83 olarak elde edilmiştir. LZW algoritmasıyla sıkıştırma yaparken mesaj uzunluğu arttıkça sıkıştırma oranının arttığı ve en yüksek sıkıştırma oranının Text-3 dosyası sıkıştırılırken elde edildiği görülmektedir. Bunun nedeni, mesaj uzunluğu arttıkça, sözlüğe eklenen yeni kelimelerin bulunma ihtimalinin artmasıdır. Yani sözlüğe eklenen 2, 3 veya daha fazla karakterli kelimelerle karşılaşıldığında, bu kelimeler yerine sözlükteki kod karşılığı kullanılacağı için mesaj boyutu arttıkça bu kelimelerle karşılaşma ihtimali de artmaktadır.

Aritmetik algoritmasıyla 56980, 111405 ve 140364 bit uzunluğunda olan orijinal mesaj dosyaları 35448, 69361 ve 88109 bit boyutuna düşürülmüştür. Sıkıştırma oranları ise %160.74, %160.62 ve %159.31 olarak hesaplanmıştır. Bu algoritmayla, mesaj boyutu arttıkça sıkıştırma oranının düştüğü ve en yüksek sıkıştırma oranının Text-1 dosyası sıkıştırılırken elde edildiği görülmektedir.

Move-to-Front algoritmasıyla 56980, 111405 ve 140364 bit uzunluğunda olan orijinal mesaj dosyaları 48840, 95490 ve 120312 bit boyutuna düşürülmüştür. Sıkıştırma oranları ise yaklaşık %116.67 olarak hesaplanmıştır. 3 metin dosyasının da sıkıştırma oranının birbirine çok yakın olduğu görülmektedir. MTF algoritması ile elde edilen sıkıştırma algoritmaları, diğer algoritmalara göre oldukça düşüktür.

Deflate algoritmasıyla 56980, 111405 ve 140364 bit uzunluğunda olan orijinal mesaj dosyaları 29496, 56832 ve 65744 bit boyutuna düşürülmüştür. Sıkıştırma oranları ise sırasıyla %193.18, %196.03 ve %213.50 olarak hesaplanmıştır. Deflate algoritması, Huffman ve LZ77 kodlamalarının birleşimiyle oluşan hibrit bir algoritma olması nedeniyle, diğer algoritmalara göre en yüksek sıkıştırma oranları bu algoritmayla elde edilmiştir. Ayrıca mesaj uzunluğu arttıkça sıkıştırma oranı da artmış ve en yüksek sıkıştırma oranı Text-3 dosyası sıkıştırılırken elde edilmiştir.

5.2. Metin Şifreleme

Önerilen veri gömme yöntemini, steganografi'nin üç temel kriterinden biri olan güvenlik konusunda daha iyi hale getirebilmek için mesaj gizlenmeden önce RSA kriptografi algoritmasıyla şifrelenmiştir. Tasarladığımız arayüzde, RSA algoritmasında bulunan genel ve özel anahtarı oluşturabilmek için gerekli olan p ve q değerlerini giriş parametresi olarak almaktayız. Seçilecek olan bu iki parametrenin etkisi test edilmiş ve sonuçlar Çizelge 5.2'de verilmiştir.

Çizelge 5.2. RSA algoritmasının giriş parametrelerine göre değişimi

Dosya Adı	Orijinal Uzunluk (bit)	p=3, q=41		p=11, q=23		p=13, q=29		p=23, q=47	
		Şifrelenmiş Uzunluk (bit)	Süre (s)	Şifrelenmiş Uzunluk (bit)	Süre (s)	Şifrelenmiş Uzunluk (bit)	Süre (s)	Şifrelenmiş Uzunluk (bit)	Süre (s)
Text-1 dosyası	56980	56980	11.88	65120	14.97	73260	15.79	89540	16.47
Text-2 dosyası	111405	111405	24.44	127320	29.76	143235	30.66	175065	32.31
Text-3 dosyası	140364	140364	30.66	160416	37.23	180468	38.95	220572	39.81

Mesaj olarak seçilen üç farklı metin dosyası, 4 ayrı (*p, q*) kombinasyonu kullanılarak RSA algoritmasıyla şifrelenmiştir. Bu testte (*p, q*) kombinasyonu her biri asal sayı olacak şekilde rastgele seçilmiştir. Çizelge 4.2'i incelediğimizde, seçilen (*p, q*) değerleri arttıkça, şifrelenmiş mesaj boyutunun da arttığı görülmektedir. Bu artışın nedeni ise, seçilen (*p, q*) değerleri arttıkça, Bölüm 3.4.1'de bahsedilen $n=p*q$ taban değeri artmaktadır. Bu da şifreleme sonucu oluşan şifreli mesajın boyutunu artırmaktadır. Bu nedenle, görüntüde oluşacak bozulmayı azaltmak amacıyla (p, q) değerlerini küçük seçmek daha avantajlı olacaktır. Ayrıca yine tabloya göre, şifreleme ve şifre çözme süresinin hem mesaj uzunluğuna hem de seçilen (*p, q*) değerine göre artmaktadır.

5.3. Test Sonuçları

Mevcut ve önerilen yöntemler 3 farklı mesaj dosyası kullanılarak 4 farklı görüntü üzerine gömülmüştür. Taşıyıcı görüntü ve gömme işlemi sonucu oluşan stego görüntüler, görüntü karşılaştırma kriterleriyle karşılaştırılmış ve sonuçlar bir sonraki bölümde verilmiştir.

Gömme işlemi için şu algoritmalar kullanılmıştır:

- Klasik 1-LSB, 2-LSB ve 3-LSB yöntemi
- Önerilen 2-LSB ve 3-LSB yöntemi
- 2. Önerilen yöntem (Önerilen 2-LSB ve 3-LSB yönteminin Deflate sıkıştırma algoritması ile birleştirilip lojistik harita tabanlı rastgele gömme yöntemi)
- 3. Önerilen yöntem (Önerilen 2-LSB ve 3-LSB yönteminin Deflate sıkıştırma algoritması ve RSA şifreleme algoritması ile birleştirilip lojistik harita tabanlı rastgele gömme yöntemi)

Gömme işlemi sırasında kullanılan giriş parametreleri ise şunlardır:

- RSA şifreleme *p değeri=3, q değeri=41*
- Lojistik harita $x_0=0.675$, $\mu=3.9763$

5.3.1. Ortalama karesel hata

Önerilen ve mevcut algoritmaların uygulaması sonucu oluşan stego görüntüler ve taşıyıcı görüntüler ortalama karesel hata (MSE) testine sokulmuş ve elde edilen sonuçlar Çizelge 5.3'de verilmiştir.

Çizelge 5.3. MSE testi sonuçları

		Klasik			Önerilen		Önerilen-2		Önerilen-3	
		1-LSB	2-LSB	3-LSB	2-LSB	3-LSB	2-LSB	3-LSB	2-LSB	3-LSB
Resim 1 "Lena"	Text-1	0.0629	0.1606	0.4475	0.0942	0.2327	0.0727	0.1966	0.1141	0.3011
	Text-2	0.1230	0.3128	0.8675	0.1849	0.4510	0.1304	0.3388	0.1924	0.4863
	Text-3	0.1547	0.3950	1,0949	0.2333	0.5666	0.1459	0.3746	0.2144	0.5322
Resim 2 "Mandrill"	Text-1	0.0122	0.0306	0.0866	0.0183	0.0447	0.0153	0.0420	0.0251	0.0686
	Text-2	0.0237	0.0600	0.1682	0.0355	0.0869	0.0290	0.0796	0.0475	0.1288
	Text-3	0.0299	0.0754	0.2098	0.0448	0.1091	0.0332	0.0909	0.0554	0.1509
Resim 3 "Peppers"	Text-1	0.0090	0.0225	0.0630	0.0135	0.0336	0.0112	0.0310	0.0183	0.0515
	Text-2	0.0175	0.0442	0.1235	0.0267	0.0656	0.0215	0.0597	0.0357	0.0981
	Text-3	0.0218	0.0558	0.1541	0.0341	0.0838	0.0246	0.0684	0.0419	0.1157
Resim 4 "Cat"	Text-1	0.0055	0.0139	0.0389	0.0085	0.0213	0.0071	0.0197	0.0114	0.0322
	Text-2	0.0107	0.0271	0.0771	0.0165	0.0421	0.0133	0.0374	0.0222	0.0617
	Text-3	0.0135	0.0341	0.0974	0.0209	0.0533	0.0154	0.0428	0.0260	0.0728

MSE, taşıyıcı ve stego görüntü pikselleri arasındaki farkın karelerinin toplamının ortalamasına eşittir. Stego görüntü ile taşıyıcı görüntü arasındaki farkın az olması istendiği için, düşük MSE değeri olması tercih edilir. Klasik 1-LSB yöntemi, görüntünün sadece son piksellerini değiştirdiği için en az bozulmanın bu yöntem ile olmuştur. Klasik 2-LSB ve 3-LSB algoritmalarıyla önerilen yöntemi kıyasladığımızda, önerilen yöntem ile daha düşük MSE değerleri elde edilmiştir. Önerilen 2. yöntemde, Deflate algoritmasıyla sıkıştırma da yapıldığı için en düşük MSE değeri bu yöntemde, 0.007075 ile Text-1 dosyası Resim-4'e gömülürken elde edilmiştir. Önerilen 3. yöntemde ise, ek olarak RSA şifreleme algoritması uygulanmış, bunun sonucunda MSE değerinin arttığı görülmüştür. Ayrıca görüntünün çözünürlüğü arttıkça veya gizlenecek mesajın boyutu azaldıkça MSE değerinin düştüğü gözlenmiştir.

5.3.2. Tepe sinyal gürültü oranı

Taşıyıcı ve Stego görüntüler Tepe Sinyal Gürültü Oranı (PSNR) testine sokulmuş ve elde edilen sonuçlar Çizelge 5.4'de verilmiştir.

Çizelge 5.4. PSNR testi sonuçları

		Klasik			Önerilen		Önerilen-2		Önerilen-3	
		1-LSB	2-LSB	3-LSB	2-LSB	3-LSB	2-LSB	3-LSB	2-LSB	3-LSB
Resim 1 "Lena"	Text-1	55.3715	51.3008	46.8514	53.6170	49.6913	54.7436	50.4237	52.7866	48.5723
	Text-2	52.4622	48.4074	43.9767	50.6893	46.8183	52.2069	48.0600	50.5169	46.4908
	Text-3	51.4637	47.3939	42.9658	49.6811	45.8272	51.7178	47.6244	50.0476	46.0986
Resim 2 "Mandrill"	Text-1	62.4823	58.5075	53.9861	60.7280	56.8595	61.5060	57.1229	59.3663	54.9992
	Text-2	59.6175	55.5756	51.1017	57.8570	53.9684	58.7355	54.3511	56.5954	52.2607
	Text-3	58.6014	54.5859	50.1419	56.8493	52.9807	58.1438	53.7763	55.9267	51.5739
Resim 3 "Peppers"	Text-1	63.8357	59.8282	55.3630	62.0569	58.0962	62.8678	58.4439	60.7352	56.2398
	Text-2	60.9356	56.9052	52.4427	59.0905	55.1898	60.0444	55.5976	57.8295	53.4450
	Text-3	59.9685	55.8915	51.4823	58.0265	54.1280	59.4566	55.0095	57.1353	52.7273
Resim 4 "Cat"	Text-1	65.9439	61.9380	57.4568	64.0891	60.0842	64.8623	60.4200	62.7730	58.2786
	Text-2	63.0601	59.0315	54.4875	61.1720	57.1166	62.1183	57.6299	59.9000	55.4555
	Text-3	62.0663	58.0370	53.4740	60.1677	56.0938	61.4934	57.0453	59.2126	54.7393

PSNR değeri, görüntüdeki tepe sinyalin gürültüye oranı olduğu için, yüksek PSNR değeri, görüntüde bozulmanın daha az olduğu anlamına gelmektedir. Elde edilen sonuçlar incelendiğinde, önerilen 2-LSB ve 3-LSB yönteminin PSNR değerlerinin klasik 2-LSB ve 3-LSB yöntemine göre daha yüksek olduğu görülmektedir. Önerilen 2. yöntemin önerilen 1. ve 3. yöntemlere göre daha iyi olduğu söylenebilir. 1-LSB yönteminden sonra en yüksek PSNR değeri, 64.86235 ile 2-LSB yöntemi ile Text-1 Resim 4' e gömülürken elde edilmiştir. Ayrıca Klasik 2-LSB ve 3-LSB yöntemine göre

en yüksek artış %10.8426 (42.96577'den 47.62437'ye) le Text-3 dosyası, Resim-1 üzerine gömülürken elde edilmiştir. Bu veriden yola çıkarak, düşük çözünürlükteki görüntüye yüksek boyuttaki veriler gömülürken, en yüksek PSNR artışının yakalanabileceği söylenebilir.

5.3.3. Ortalama fark

Taşıyıcı ve Stego görüntüler, Ortalama Fark (AD) testine tabi tutulmuş ve sonuçları Çizelge 5.5'de verilmiştir.

Çizelge 5.5. Ortalama fark testi sonuçları

		Klasik			Önerilen		Önerilen-2		Önerilen-3	
		1-LSB	2-LSB	3-LSB	2-LSB	3-LSB	2-LSB	3-LSB	2-LSB	3-LSB
Resim 1 "Lena"	Text-1	0.0625	0.0793	0.1108	0.0627	0.0840	0.0443	0.0666	0.0703	0.1030
	Text-2	0.1226	0.1548	0.2155	0.1228	0.1634	0.0807	0.1167	0.1224	0.1708
	Text-3	0.1544	0.1953	0.2716	0.1548	0.2055	0.0907	0.1298	0.1379	0.1891
Resim 2 "Mandrill"	Text-1	0.0121	0.0152	0.0214	0.0121	0.0162	0.0092	0.0141	0.0151	0.0230
	Text-2	0.0236	0.0297	0.0417	0.0236	0.0316	0.0175	0.0267	0.0288	0.0435
	Text-3	0.0298	0.0375	0.0522	0.0298	0.0396	0.0200	0.0305	0.0336	0.0509
Resim 3 "Peppers"	Text-1	0.0089	0.0112	0.0156	0.0089	0.0119	0.0067	0.0103	0.0110	0.0171
	Text-2	0.0174	0.0218	0.0305	0.0174	0.0232	0.0129	0.0199	0.0215	0.0327
	Text-3	0.0218	0.0275	0.0382	0.0220	0.0294	0.0148	0.0228	0.0252	0.0385
Resim 4 "Cat"	Text-1	0.0055	0.0069	0.0096	0.0055	0.0074	0.0042	0.0066	0.0069	0.0107
	Text-2	0.0107	0.0134	0.0190	0.0108	0.0146	0.0080	0.0125	0.0134	0.0207
	Text-3	0.0134	0.0169	0.0240	0.0136	0.0185	0.0092	0.0143	0.0157	0.0243

Ortalama Fark, taşıyıcı ve stego görüntü pikselleri arasındaki farkın ortalamasına eşittir. Stego görüntünün taşıyıcı görüntüye benzer olması istenildiğinden, ortalama fark değerinin küçük olması beklenir. Elde edilen sonuçlar incelendiğinde, en küçük ortalama fark değeri olan 0.042, önerilen 2. 2-LSB algoritması ile Text-1 mesajının Resim-4 üzerine gömülmesi sonucu elde edilmiştir. Ayrıca, önerilen 2. yöntemin, diğer klasik ve önerilen yöntemlere göre daha üstün olduğu görülmektedir. Bununla birlikte, resmin çözünürlüğü arttıkça veya gizlenecek mesajın uzunluğu azaldıkça, ortalama fark değerinin azaldığı da görülmektedir.

5.3.4. Yapısal benzerlik testi

Yapısal benzerlik testi (SSIM), taşıyıcı ve stego görüntülerin birbirine yapısal olarak ne kadar benzediğini gösteren testtir. 0 ile 1 arasında değerler almakta ve 0 görüntülerin birbirinden tamamen farklı, 1 ise görüntülerin özdeş olduğunu ifade etmektedir. Bu tezde elde edilen stego ve taşıyıcı görüntüler yapısal benzerlik testine sokulmuş ve elde edilen sonuçlar Çizelge 5.6'de verilmiştir.

Elde edilen sonuçlar incelendiğinde, bazı sonuçların 1 çıktığı görülmektedir. Virgülden sonra 4 haneye kadar yapılan yuvarlama sonucu oluşan bu durum görüntülerin özdeş olduğu anlamına gelmemekte fakat birbirine çok benzer olduğu anlamına gelmektedir. Önerilen 2. yöntemde elde edilen sonuçların, hem diğer önerilen yöntemlere hem de klasik yöntemlere göre daha yüksek olduğu görülebilmektedir.

Ayrıca, tablo genel olarak incelendiği zaman elde edilen yapısal benzerlik değerlerin tamamı 0.99'un üzerinde olduğu görülmektedir. Bu nedenle yapısal benzerlik testinin steganografi'de kullanılmak için uygun bir kriter olmadığı sonucu çıkarılabilir.

Çizelge 5.6. SSIM testi sonuçları

		Klasik			Önerilen		Önerilen-2		Önerilen-3	
		1-LSB	2-LSB	3-LSB	2-LSB	3-LSB	2-LSB	3-LSB	2-LSB	3-LSB
Resim 1 "Lena"	Text-1	0.9999	0.9997	0.9994	0.9998	0.9997	0.9998	0.9996	0.9997	0.9994
	Text-2	0.9997	0.9993	0.9982	0.9996	0.9991	0.9995	0.9990	0.9997	0.9993
	Text-3	0.9996	0.9991	0.9977	0.9995	0.9988	0.9996	0.9992	0.9995	0.9989
Resim 2 "Mandrill"	Text-1	1.0000	1.0000	0.9999	1.0000	1.0000	1.0000	0.9999	1.0000	0.9999
	Text-2	1.0000	0.9999	0.9998	1.0000	0.9999	1.0000	0.9999	0.9999	0.9998
	Text-3	0.9999	0.9999	0.9998	0.9999	0.9999	0.9999	0.9999	0.9999	0.9998
Resim 3 "Peppers"	Text-1	1.0000	1.0000	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Text-2	1.0000	0.9999	0.9998	1.0000	0.9999	1.0000	0.9999	1.0000	0.9999
	Text-3	1.0000	0.9999	0.9998	1.0000	0.9999	1.0000	0.9999	1.0000	0.9999
Resim 4 "Cat"	Text-1	0.9999	0.9998	0.9995	0.9999	0.9997	0.9999	0.9998	0.9999	0.9997
	Text-2	0.9999	0.9996	0.9990	0.9998	0.9994	0.9999	0.9997	0.9998	0.9995
	Text-3	0.9998	0.9996	0.9987	0.9997	0.9993	0.9999	0.9996	0.9998	0.9994

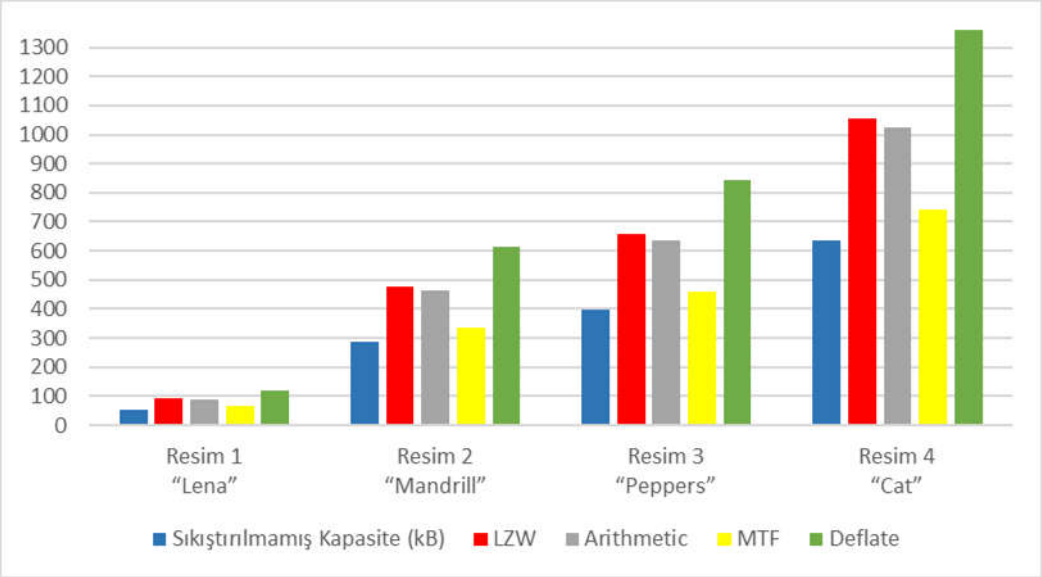
5.4. Kapasite Testi

Kapasite, bir görüntü üzerine gizlenebilecek maksimum veri miktarı şeklinde tanımlanabilir. Kapasite, steganografi'nin üç temel kriterinden bir tanesidir. Bu nedenle steganografi yöntemleri karşılaştırılırken, görüntülerin kapasitelerine de bakılmaktadır.

Çizelge 5.7. Taşıyıcı görüntülerin tahmini kapasiteleri

	Sıkıştırılmamış Kapasite (kB)	LZW	Arithmetic	MTF	Deflate
Resim 1 "Lena"	55.62	92.23	89.40	64.89	118.75
Resim 2 "Mandrill"	288.00	477.59	462.93	336.01	614.88
Resim 3 "Peppers"	395.51	655.87	635.74	461.44	844.41
Resim 4 "Cat"	635.98	1054.64	1022.27	741.99	1357.81

Çizelge 5.7'de önerilen yöntemde kullanılan veri sıkıştırma algoritmalarının görüntü kapasitesi üzerinde etkisi görülmektedir. Burada verilen kapasite değerleri tahmini olup, Çizelge 5.1'de hesaplanan sıkıştırma oranlarına bakılarak hesaplanmıştır. Ayrıca sıkıştırma yöntemi olarak 3-LSB yöntemi kullanılmıştır. Buna göre, en yüksek kapasite artışı, Deflate algoritması ile elde edilmiştir. Bu algoritma ile 635.98kB veri kapasitesi olan Resim-4'ün kapasitesi 1357.81 kB'a yükseltilmiş, %113.5'lik bir artış yakalanmıştır. Şekil 5.2'de 3-LSB yöntemi ve belirtilen sıkıştırma algoritması kullanılarak elde edilen kapasite değerleri grafik olarak verilmiştir.



Şekil 5.2. Kullanılan sıkıştırma algoritmasının görüntü kapasitesine etkisi

5.5. Görsel Analiz

Steganografi'nin temel amacı şüphe uyandırmamaktadır. Bu nedenle hangi gömme algoritma kullanılırsa kullanılsın elde edilen stego görüntülerin 3. kişilerin eline geçmesi durumunda şüphe uyandırmaması gereklidir. Bu nedenle stego görüntüde insan görme sisteminin fark edebileceği değişiklikler bozulmalar olmaması gereklidir.

Şekil 5.3'de önerilen 2-LSB yöntemiyle, Şekil 5.4'de ise önerilen 3-LSB yöntemiyle her bir görüntü üzerine 17.3 kB veri gömülerek elde edilen stego görüntüler verilmiştir. Stego görüntüleri orijinal taşıyıcı görüntüler ile kıyasladığımızda görsel olarak herhangi bir deformasyon görülmemektedir.

a) Lena

b) Mandril

c) Cat

d) Peppers

Şekil 5.3. Önerilen 2-LSB yöntemi uygulanarak elde edilen stego görüntüler

a) Lena

b) Mandril

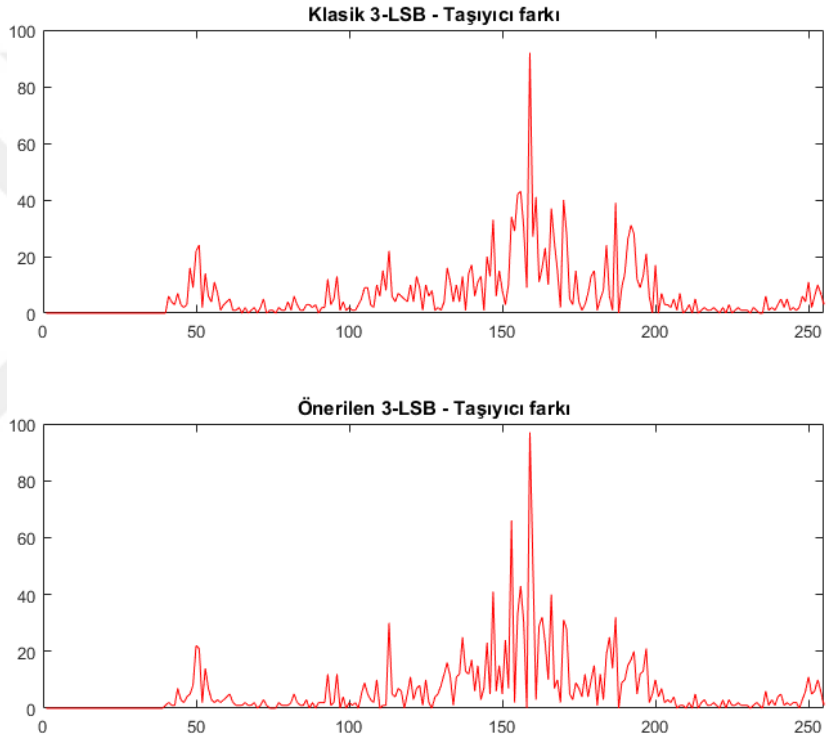
c) Cat

d) Peppers

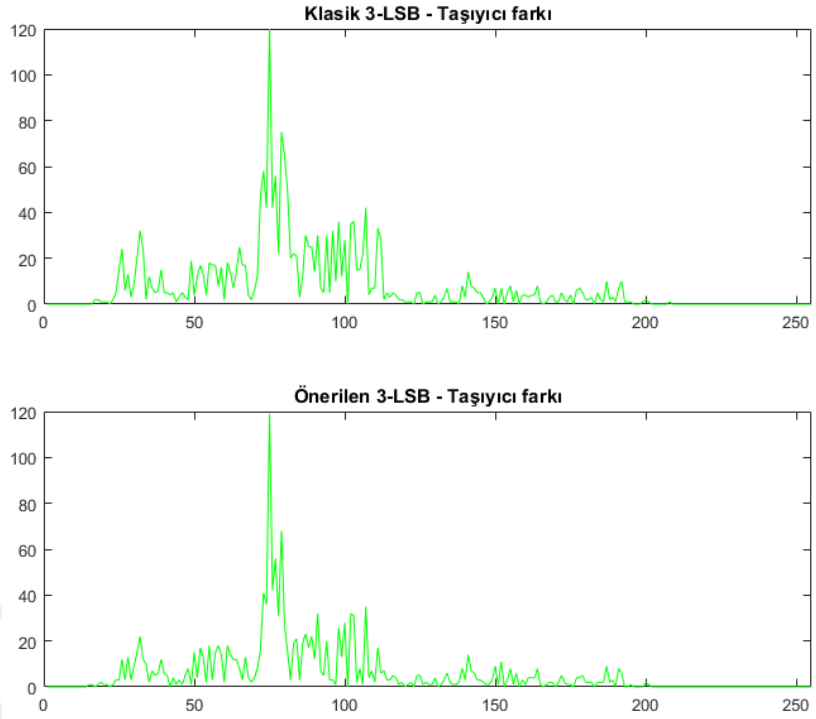
Şekil 5.4. Önerilen 3-LSB yöntemi uygulanarak elde edilen stego görüntüler

5.6. Histogram Analizi

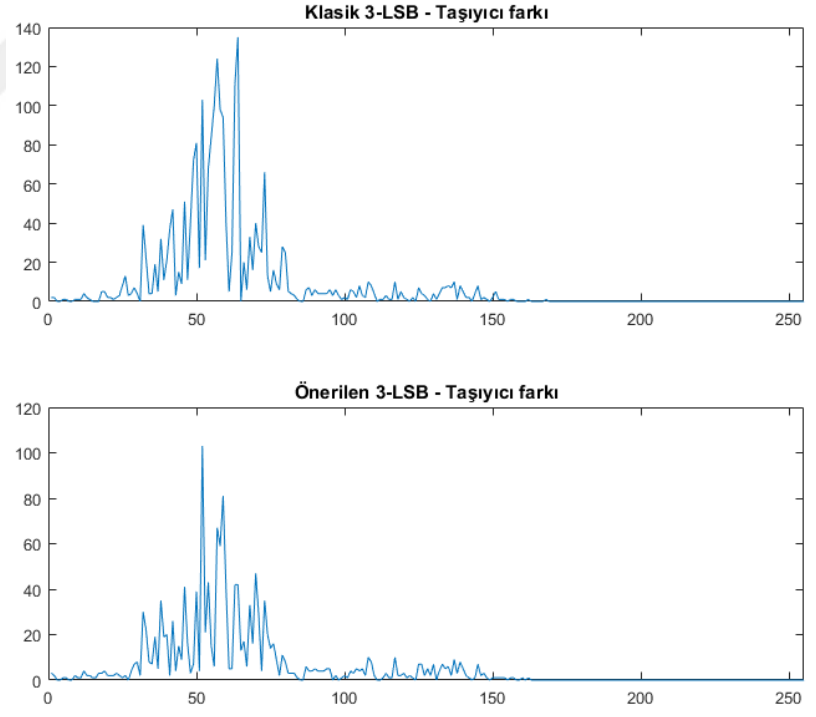
Önerilen 3-LSB yöntemi ile klasik 3-LSB yöntemi kullanarak 225x225 piksel çözünürlüğündeki ‘lena’ görüntüsüne 17.3 kB veri gömülmüş ve elde edilen stego görüntüler ile taşıyıcı görüntülerin histogram farkları çıkartılmıştır. İki yöntemde de gömme işleminden önce şifreleme ve sıkıştırma algoritması uygulanmamıştır. Şekil 5.5’de R kanalının histogram farkları, Şekil 5.6’da G kanalının histogram farkları ve Şekil 5.7’de B kanalının histogram farkları verilmiştir. Şekiller incelendiği zaman, özellik yeşil ve mavi kanallarda, önerilen yöntemde klasik yönteme göre histogram farkında belirgin bir azalma olmuştur.



Şekil 5.5. Kırmızı (R) kanalın histogram farkları



Şekil 5.6. Yeşil (G) kanalın histogram farkları



Şekil 5.7. Mavi (B) kanalın histogram farkları

6. SONUÇLAR VE ÖNERİLER

6.1 Sonuçlar

Bu tezde, bilgi güvenliği yöntemlerinden bir tanesi olan steganografi yöntemleri incelenmiş ve görüntü steganografisinde yeni bir yöntem geliştirilmiştir. Önerdiğimiz bu yöntem, LSB değiştirme yöntemine dayanmakta olup, LSB değiştirme yöntemi uygulanması sonucu taşıyıcı ile stego görüntü arasında oluşan piksel farklarını azaltmaya çalışmaktadır. Bu sayede, steganografi'nin üç temel kriterinden biri olan algılanamazlık kriterinde iyileşme sağlanmış, yapılan testlerle bu durum doğrulanmıştır.

Önerdiğimiz yöntemin LSB değiştirme yöntemine dayalı olması nedeniyle, 3. kişiler tarafından gizli mesajın çıkarılabilmesi mümkündür. Bu problemi çözmek için, mesaj bitlerini taşıyıcı görüntü içerisine sıralı gömmek yerine kaos tabanlı rastgele sayı üreteci kullanarak rastgele gömülmüştür. Ayrıca, önerilen yöntemi güvenlik/dayanıklılık kriterinde iyileştirmek için, RSA şifreleme algoritması kullanarak mesaj gizlenmeden önce şifrelenmiştir. Daha sonra, hem 3. kriter olan kapasite kriterinde iyileştirme sağlamak hem de RSA şifreleme algoritmasının kullanılması sonucu oluşacak veri boyutu artışını telafi etmek için veri sıkıştırma teknikleri önerilen yöntem ile birleştirilmiştir. Sıkıştırma tekniği olarak LZW, Arithmetic, MTF ve Deflate olmak üzere 4 farklı yöntem uygulanmış, bu yöntemler arasından en iyi sıkıştırma oranı Deflate algoritması ile elde edilmiştir. Bu nedenle, mesaj taşıyıcı içine gizlenmeden önce Deflate algoritması ile sıkıştırılmıştır.

Önerilen bu yöntemleri ve mevcut LSB değiştirme yöntemini daha kolay uygulayabilmek için MATLAB GUI ortamında arayüz tasarlanmıştır.

Önerilen bu yöntemleri ve mevcut LSB değiştirme yöntemi, farklı çözünürlükte 4 taşıyıcı görüntü üzerine farklı boyutlarda 3 mesaj dosyası gizleyerek test edilmiştir. Elde edilen stego görüntüler ile taşıyıcı görüntüler görüntü kalitesi değerlendirme kriterlerine göre karşılaştırıldığında en yüksek PSNR değeri 64.86 dB olarak, en yüksek PSNR artış oranı ise %10.84 olarak elde edilmiştir. Önerilen yöntemin farklı kombinasyonlarının tamamında PSNR değeri klasik LSB yöntemine göre daha yüksek çıkmıştır. Ayrıca önerilen yöntemde Deflate sıkıştırma algoritmasının kullanılması sonucu, taşıyıcı görüntülerin gizleme kapasitelerinde %113.5'a kadar artış sağlanmıştır.

6.2 Öneriler

Bu tez çalışmasında, görüntü steganografisinde uygulanmak üzere yeni bir yöntem önerilmiştir. Önerilen yöntem, uzamsal yani görüntü düzleminde uygulanmış, frekans düzleminde herhangi bir çalışma yapılmamıştır. Bu nedenle, önerdiğimiz yöntemi farklı dönüşümler kullanarak frekans düzlemine adapte etmeye yönelik yeni çalışmalar yapılabilir. Bu çalışmada uygulanmayan, farklı sıkıştırma teknikleri de uygulanarak kapasite de daha da yüksek artış oranı elde edilmeye yönelik çalışmalar yapılabilir. Ayrıca önerilen yöntem mevcut başka yöntemlerle birleştirip daha iyi sonuçlar elde edilebilir.

KAYNAKLAR

Akhtar, N., Johri, P. ve Khan, S., 2013, Enhancing the Security and Quality of LSB Based Image Steganography, *2013 5th International Conference and Computational Intelligence and Communication Networks*, 385-390.

Akhtar, N., Khan, S. ve Johri, P., 2014, An improved inverted LSB image steganography, *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, 749-755.

Akhtar, N., Ahamad, V. ve Javed, H., 2017, A compressed LSB steganography method, *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, 1-7.

Arora, A., Singh, M. P., Thakral, P. ve Jarwal, N., 2016, Image steganography using enhanced LSB substitution technique, *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 386-389.

Bacon, F., 1640, Of the Advancement and Proficiencie of Learning or the Partitions of Sciences, London, T. Williams, p.

Bedi, P., Bansal, R. ve Sehgal, P., 2011, Using PSO in Image Hiding Scheme Based on LSB Substitution, In: Advances in Computing and Communications: First International Conference, ACC 2011, Kochi, India, July 22-24, 2011, Proceedings, Part III, Eds: Abraham, A., Mauri, J. L., Buford, J. F., Suzuki, J. ve Thampi, S. M., *Berlin, Heidelberg*: Springer Berlin Heidelberg, p. 259-268.

Bedwal, T. ve Kumar, M., 2013, An enhanced and secure image steganographic technique using RGB-box mapping, *Confluence 2013: The Next Generation Information Technology Summit (4th International Conference)*, 385-393.

Chang, C.-C., Hsiao, J.-Y. ve Chan, C.-S., 2003, Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy, *Pattern Recognition*, 36 (7), 1583-1595.

Chen, S.-K., 2011, A module-based LSB substitution method with lossless secret data compression, *Computer Standards & Interfaces*, 33 (4), 367-371.

Chikouche, S. L. ve Chikouche, N., 2017, An improved approach for lsb-based image steganography using AES algorithm, *2017 5th International Conference on Electrical Engineering - Boumerdes (ICEE-B)*, 1-6.

Clelland, C. T., Risca, V. ve Bancroft, C., 1999, Hiding messages in DNA microdots, *Nature*, 399, 533.

Darabkh, K. A., Jafar, I. F., Al-Zubi, R. T. ve Hawa, M., 2014, An improved image least significant bit replacement method, *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1182-1186.

Devi, A., Sharma, A. ve Rangra, A., 2015, A Review on DES, AES and Blowfish for Image Encryption & Decryption, *International Journal of Computer Science and Information Technologies*, 6 (3), 3034-3036.

Dongardive, P., Gupta, N. ve Barde, C., 2015, Improved security color grace steganography with grace to text encoding and LSB, *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, 1-6.

Goyal, S., Ramaiya, M. ve Dubey, D., 2015, Improved Detection of 1-2-4 LSB Steganography and RSA Cryptography in Color and Grayscale Images, *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, 1120-1124.

Hemani, S. S., 2017, A Survey of Digital Watermarking Techniques and Performance Evaluation Metrics, *International Journal of Engineering Trends and Technology (IJETT)*, 46 (2), 128-132.

Islam, A. U., Khalid, F., Shah, M., Khan, Z., Mahmood, T., Khan, A., Ali, U. ve Naeem, M., 2016, An improved image steganography technique based on MSB using bit differencing, *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, 265-269.

Islam, M. R., Siddiqa, A., Md. Palash, U., Mandal, A. K. ve Hossain, M. D., 2014, An efficient filtering based approach improving LSB image steganography using status bit along with AES cryptography, *2014 International Conference on Informatics, Electronics & Vision (ICIEV)*, 1-6.

J. Brassil , S. L., N. F. Maxemchuk , L. O'Gorman, 1994, Hiding information in document images, *In Proceedings of the Conference on Information Sciences and Systems*, Baltimore, 482-489.

Jacqueline Tobin, R. G. D., Maude S. Wahlman 1999, Hidden in Plain View: The Secret Story of Quilts and the

Underground Railroad, *New York*, Doubleday, p.

Johnson, D., 2008, White King and Red Queen: How the Cold War Was Fought on the Chessboard, *New York*, Houghton Mifflin Company, p.

Juneja, M. ve Sandhu, P. S., 2013, A Review of Cryptography Techniques and Implementation of AES for Images, *International Journal of Computer Science and Electronics Engineering (IJCSEE)*, 1 (4), 478-482.

Kalita, M. ve Tuithung, T., 2016, A novel steganographic method using 8-neighboring PVD (8nPVD) and LSB substitution, *2016 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 1-5.

Karakış, R. ve Güler, İ., 2014, An application of fuzzy logic-based image steganography, *2014 22nd Signal Processing and Communications Applications Conference (SIU)*, 156-159.

Khodaei, M. ve Faez, K., 2010, Image Hiding by Using Genetic Algorithm and LSB Substitution, In: Image and Signal Processing: 4th International Conference, ICISP 2010, Trois-Rivières, QC, Canada, June 30-July 2, 2010. Proceedings, Eds: Elmoataz, A., Lezoray, O., Nouboud, F., Mammass, D. ve Meunier, J., *Berlin, Heidelberg*: Springer Berlin Heidelberg, p. 404-411.

Khodaei, M. ve Faez, K., 2012, New adaptive steganographic method using least significant- bit substitution and pixel-value differencing, *IET Image Processing*, 6 (6), 677-686.

Kolata, G., 1998, The New York Times, A mystery unraveled, twice F1-F6.

Langdon, G. G., 1984, An Introduction to Arithmetic Coding, *IBM Journal of Research and Development*, 28 (2), 135-149.

Manjula, Y. ve Shivakumar, K. B., 2016, Enhanced secure image steganography using double encryption algorithms, *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 705-708.

Meunier, J., 2000, Why a Brewster angle microscope?, *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 171 (1), 33-40.

Mohan Durvey, D. S., 2014, A Review Paper on Digital Watermarking, *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 3 (4), 99-105.

Mousa, A. ve Hamad, A., 2006, Evaluation of the RC4 Algorithm for Data Encryption, *International Journal of Computer Science & Applications*, 3 (2), 44-56.

Nag, A., Ghosh, S., Biswas, S., Sarkar, D. ve Sarkar, P. P., 2012, An image steganography technique using X-box mapping, *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012)*, 709-713.

Nickfarjam, A. M., Ebrahimpour-komleh, H. ve Najafabadi, A. P., 2014, Image hiding using neighborhood similarity, *2014 6th Conference on Information and Knowledge Technology (IKT)*, 79-82.

Pasco, R. C., 1976, Source coding algorithms for fast data compression, *Stanford University*.

Patil, S. ve Katariya, S. S., 2015, Data Hiding Techniques: A Review, *International Journal of Computer Applications*, 122 (17), 1-3.

Rajput, V., Tiwari, S. K. ve Gupta, R., 2016, An enhanced image security using improved RSA cryptography and spatial orientation tree compression method, *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 327-331.

Reeds, J., 1998, Solved: The Ciphers In Book III Of Trithemius's Steganographia, *Cryptologia*, 22 (4), 291-317.

Rissanen, J. ve Langdon, G. G., 1979, Arithmetic Coding, *IBM Journal of Research and Development*, 23 (2), 149-162.

Schyndel, R. G. v., Tirkel, A. Z. ve Osborne, C. F., 1994, A digital watermark, *Proceedings of 1st International Conference on Image Processing*, 86-90 vol.82.

Shanmugasundaram, S. ve Lourdusamy, R., 2011, A Comparative Study Of Text Compression Algorithms, *International Journal of Wisdom Based Computing*, 1 (3), 68-76.

Shimanovsky, B., Feng, J. ve Potkonjak, M., 2003, Hiding Data in DNA, In: Information Hiding: 5th International Workshop, IH 2002 Noordwijkerhout, The Netherlands, October 7-9, 2002 Revised Papers, Eds: Petitcolas, F. A. P., *Berlin, Heidelberg*: Springer Berlin Heidelberg, p. 373-386.

Singh, A. ve Singh, H., 2015, An improved LSB based image steganography technique for RGB images, *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 1-4.

Sugathan, S., 2016, An improved LSB embedding technique for image steganography, *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 609-612.

Tactician, A. t., 1990, How to Survive Under Siege/ Aineias the Tactician, In: Oxford: Clarendon Ancient History Series, Eds, p.

Tahghighi, M., Mousavi, M. ve Khadivi, P., 2010, Hardware implementation of a novel adaptive version of Deflate compression algorithm, *2010 18th Iranian Conference on Electrical Engineering*, 566-569.

Thien, C.-C. ve Lin, J.-C., 2003, A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function, *Pattern Recognition*, 36 (12), 2875-2881.

Welch, T. A., 1984, A Technique for High-Performance Data Compression, *Computer*, 17 (6), 8-19.

Wilkins, E. H., 1974, A History of Italian Literature, Harvard University Press, p.

Wu, H. C., Wu, N. I., Tsai, C. S. ve Hwang, M. S., 2005, Image steganographic scheme based on pixel-value differencing and LSB replacement methods, *IEE Proceedings - Vision, Image and Signal Processing*, 152 (5), 611-615.

Zhou, X., Gong, W., Fu, W. ve Jin, L., 2016, An improved method for LSB based color image steganography combined with cryptography, *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 1-4.

EKLER

EK-1 Önerilen 2-bit gömme algoritmasının Matlab kodu

```
function [StegoImage BinaryMesajLenght] = ProposedLSB_2(CoverImage,
CompressedMessage,EmbeddingOrder,Chaos_mu, Chaos_Xzero,maks,size2,lgt)
%Resim Girişi ve boyut hesabı
InputImageColor=CoverImage;
[m n k] =size(InputImageColor);
%Mesaj girilmesi ve binary formatına dönüştüme
Mesaj_X=dec2bin(CompressedMessage);
KarakterUzunlugu=size(Mesaj_X,2);% Herbir mesaj karakterinin kaç bitten oluştuğunu
gösteren değişken..
MesajUzunlugu=max(size(CompressedMessage));
BinaryMesaj=Mesaj_X(1,:);
[x y]=size(Mesaj_X);
for i=2:x
   BinaryMesaj=[BinaryMesaj Mesaj_X(i,:)]; %binary mesajı arka arkaya birleştirilir.
end
BinaryMesajLenght=max(size(BinaryMesaj));
%Gömme işlemi kaos ile mi yapılacak yoksa sıralı mı yapılacak onun seçimi
%(guiden geliyor)
if EmbeddingOrder == 1
   GommeSirasi =linspace(61,BinaryMesajLenght+60,BinaryMesajLenght);
elseif EmbeddingOrder == 2
%Kaos üreteci ile gömme yerlerinin belirlenmesi
GommeSirasi=kaos_gen(BinaryMesajLenght,Chaos_Xzero,Chaos_mu,m*n*k);
end
 %Gömme İşlemi
Capacity=m*n*3; %Taşıyıcı görüntünün kapasitesi yeterlimi diye bakılıyor
if Capacity<BinaryMesajLenght
   display('*Mesaj Kapasiteden Büyük');
   Sonuc=[0 0 0 0 0 0 0 0 0 0];
   return
end
x=uint8(fix(BinaryMesajLenght/(3*n)));
y=ceil(mod(BinaryMesajLenght/3,n));
OutputImageColor=InputImageColor;
for i=1:m
   for j=1:n
      for RGB_Secim=1:3
         for Bit_Index=1:2
            BulunacakSayi=((((i-1)*n)+(j-1))*3+(RGB_Secim-1))*2+Bit_Index;
            BulunanYer=find(GommeSirasi==BulunacakSayi);
            if isempty(BulunanYer)
            else
               index=BulunanYer;
               A=bitget(InputImageColor(i,j,RGB_Secim),Bit_Index);
               B=uint8(BinaryMesaj(index))-48;
                  if Bit_Index==1
                     if A==B
```

```
                    end
                    if A>B
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)-1;
                    end
                    if A<B
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)+1;
                    end
                end
                if Bit_Index==2
                    if A==B
                    end
                    if A>B
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)-2;
                    end
                    if A<B
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)+2;
                    end
                end
            end
        end
    end
  end
end
%Başlangıç parametreleri yazılıyor
OutputImageColor=BaslangicParametreleriYaz(OutputImageColor,KarakterUzunlugu,
MesajUzunlugu,maks,size2,lgt);
index=1;    %Başlangıç parametreleri bozulmasın diye
for i=1:m
   for j=1:n
      for RGB_Secim=1:3
         Fark=int8(int16(InputImageColor(i,j,RGB_Secim))-
int16(OutputImageColor(i,j,RGB_Secim)));
         if index>60 %Başlangıç parametreleri bozulmasın diye
            if Fark==3
               if OutputImageColor(i,j,RGB_Secim)<=251
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)+4;
               end
            end
            if Fark==-3
               if OutputImageColor(i,j,RGB_Secim)>=4
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)-4;
               end
            end
         end
         index=index+1;
      end
   end
end

StegoImage = OutputImageColor;
```

EK-2 Önerilen 3-bit gömme algoritmasının Matlab kodu

```
function [StegoImage BinaryMesajLenght] =
ProposedLSB_3(CoverImage,CompressedMessage,EmbeddingOrder,Chaos_mu,
Chaos_Xzero,maks,size2,lgt)
%Resim Girişi ve boyut hesabı
InputImageColor=CoverImage;
[m n k] =size(InputImageColor);
%Mesaj girilmesi ve binary formatına dönüştüme
Mesaj_X=dec2bin(CompressedMessage);
KarakterUzunlugu=size(Mesaj_X,2);% Herbir mesaj karakterinin kaç bitten oluştuğunu
gösteren değişken..
MesajUzunlugu=max(size(CompressedMessage));
BinaryMesaj=Mesaj_X(1,:);
[x y]=size(Mesaj_X);
for i=2:x
   BinaryMesaj=[BinaryMesaj Mesaj_X(i,:)]; %binary mesajı arka arkaya birleştirilir.
end
BinaryMesajLenght=max(size(BinaryMesaj));
%Gömme işlemi kaos ile mi yapılacak yoksaa sıralı mı yapılacak onun seçimi
%(guiden geliyor)
if EmbeddingOrder == 1
   GommeSirasi =linspace(61,BinaryMesajLenght+60,BinaryMesajLenght);
elseif EmbeddingOrder == 2
%Kaos üreteci ile gömme yerlerinin belirlenmesi
GommeSirasi=kaos_gen(BinaryMesajLenght,Chaos_Xzero,Chaos_mu,m*n*k);
end
 %Gömme İşlemi
Capacity=m*n*3;
if Capacity<BinaryMesajLenght
   display('*Mesaj Kapasiteden Büyük');
   Sonuc=[0 0 0 0 0 0 0 0 0 0];
   return
end
x=uint8(fix(BinaryMesajLenght/(3*n)));
y=ceil(mod(BinaryMesajLenght/3,n));
OutputImageColor=InputImageColor;
for i=1:m
   for j=1:n
      for RGB_Secim=1:3
         for Bit_Index=1:3
            BulunacakSayi=((((i-1)*n)+(j-1))*3+(RGB_Secim-1))*3+Bit_Index;
            BulunanYer=find(GommeSirasi==BulunacakSayi);
            if isempty(BulunanYer)
            else
               index=BulunanYer;
               A=bitget(InputImageColor(i,j,RGB_Secim),Bit_Index);
               B=uint8(BinaryMesaj(index))-48;

                  if Bit_Index==1
                     if A==B
```

```
                    end
                    if A>B
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)-1;
                    end
                    if A<B
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)+1;
                    end
                end
                if Bit_Index==2
                    if A==B
                    end
                    if A>B
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)-2;
                    end
                    if A<B
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)+2;
                    end
                end
                if Bit_Index==3
                    if A==B
                    end
                    if A>B
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)-4;
                    end
                    if A<B
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)+4;
                    end
                end
            end
        end
    end
end
%Başlangıç parametreleri yazılıyor
OutputImageColor=BaslangicParametreleriYaz(OutputImageColor,KarakterUzunlugu,
MesajUzunlugu,maks,size2,lgt);
index=1;%Başlangıç parametreleri bozulmasın diye
for i=1:m
    for j=1:n
        for RGB_Secim=1:3
            Fark=int8(int16(InputImageColor(i,j,RGB_Secim))-
int16(OutputImageColor(i,j,RGB_Secim)));
            if index>60 %Başlangıç parametreleri bozulmasın diye
                if Fark>=5
                    if OutputImageColor(i,j,RGB_Secim)<=247
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)+8;
                    end
                end
                if Fark<=-5
                    if OutputImageColor(i,j,RGB_Secim)>=8
```

```
OutputImageColor(i,j,RGB_Secim)=OutputImageColor(i,j,RGB_Secim)-8;
                end
              end
            end
            index=index+1;
        end
    end
end

StegoImage = OutputImageColor;
```

EK-3 Önerilen 2-bit çıkarma algoritmasının Matlab kodu

```
function [ExtractedMessage,maks,size2,lgt] =
ex_ProposedLSB_2(StegoImage,EmbeddingOrder,Chaos_mu, Chaos_Xzero);
%Resim Girişi ve boyut hesabı
StegoImageColor=StegoImage;
%InputImageColor=imread('LenaRenkli.bmp');
[m n k] =size(StegoImageColor);

%Başlangıç değerleri okunuyor..
[KarakterUzunlugu,MesajUzunlugu,maks,size2,lgt]=
BaslangicParametreleriOku(StegoImageColor);
%Gömme işlemi kaos ile mi yapılacak yoksa sıralı mı yapılacak onun seçimi
%(guiden geliyor)
if EmbeddingOrder == 1
   GommeSirasi
=linspace(61,double(MesajUzunlugu*KarakterUzunlugu)+60,double(MesajUzunlugu*
KarakterUzunlugu));
elseif EmbeddingOrder == 2
%Kaos üreteci ile gömme yerlerinin belirlenmesi
GommeSirasi=kaos_gen(MesajUzunlugu*KarakterUzunlugu,Chaos_Xzero,Chaos_mu,
m*n*k);
end
 BinaryMesaj=zeros(1,MesajUzunlugu*KarakterUzunlugu);

for i=1:m
   for j=1:n
      for RGB_Secim=1:3
         for Bit_Index=1:2
            BulunacakSayi=((((i-1)*n)+(j-1))*3+(RGB_Secim-1))*2+Bit_Index;
            BulunanYer=find(GommeSirasi==BulunacakSayi);
            if isempty(BulunanYer)
            else
               BinaryMesaj(BulunanYer)=
bitget(StegoImageColor(i,j,RGB_Secim),Bit_Index);
            end
         end
      end
   end
end
Mesaj_X=reshape(BinaryMesaj,KarakterUzunlugu,[])';
Mesaj_X=int16(Mesaj_X);
for i=1:max(size(Mesaj_X))
Mesaj(i)=bi2de(Mesaj_X(i,:),'left-msb');
end
ExtractedMessage=uint16(Mesaj);
```

EK-4 Önerilen 3-bit çıkarma algoritmasının Matlab kodu

```
function [ExtractedMessage,maks,size2,lgt] =
ex_ProposedLSB_3(StegoImage,EmbeddingOrder,Chaos_mu, Chaos_Xzero);
%Resim Girişi ve boyut hesabı
StegoImageColor=StegoImage;
%InputImageColor=imread('LenaRenkli.bmp');
[m n k] =size(StegoImageColor);
%Başlangıç değerleri okunuyor..
[KarakterUzunlugu,MesajUzunlugu,maks,size2,lgt]=
BaslangicParametreleriOku(StegoImageColor);

%Gömme işlemi kaos ile mi yapılacak yoksaa sıralı mı yapılacak onun seçimi
%(guiden geliyor)
if EmbeddingOrder == 1
   GommeSirasi
=linspace(61,double(MesajUzunlugu*KarakterUzunlugu)+60,double(MesajUzunlugu*
KarakterUzunlugu));
elseif EmbeddingOrder == 2
%Kaos üreteci ile gömme yerlerinin belirlenmesi
GommeSirasi=kaos_gen(MesajUzunlugu*KarakterUzunlugu,Chaos_Xzero,Chaos_mu,
m*n*k);
end
BinaryMesaj=zeros(1,MesajUzunlugu*KarakterUzunlugu);

for i=1:m
   for j=1:n
      for RGB_Secim=1:3
         for Bit_Index=1:3
            BulunacakSayi=((((i-1)*n)+(j-1))*3+(RGB_Secim-1))*3+Bit_Index;
            BulunanYer=find(GommeSirasi==BulunacakSayi);
            if isempty(BulunanYer)
            else
               BinaryMesaj(BulunanYer)=
bitget(StegoImageColor(i,j,RGB_Secim),Bit_Index);
            end
         end
      end
   end
end
Mesaj_X=reshape(BinaryMesaj,KarakterUzunlugu,[])';
Mesaj_X=int16(Mesaj_X);
for i=1:max(size(Mesaj_X))
Mesaj(i)=bi2de(Mesaj_X(i,:),'left-msb');
end
ExtractedMessage=uint16(Mesaj);
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Özcan ÇATALTAŞ
Uyruğu	**:**	T.C.
Doğum Yeri ve Tarihi	**:**	Konya, 16.11.1992
Telefon	**:**	0332 223 33 79
Faks	**:**	
e-mail	**:**	ozcancataltas@selcuk.edu.tr

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	Dolapoğlu Anadolu Lisesi, Selçuklu, Konya	2010
Üniversite	:	Mevlana Üniversitesi, Selçuklu, Konya	2014
Yüksek Lisans	:	Selçuk Üniversitesi	2018

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2014 - 2014	Endüstriyel Elektrik Elektronik SAN.	Arge Mühendisi
2014 - …	S.Ü. Teknoloji Fakültesi	Arş. Gör.

UZMANLIK ALANI

- Steganografi
- Yapay Zeka

YABANCI DİLLER

- İngilizce – YDS: 87.5

YAYINLAR

Cataltas O., Tutuncu K., “Implementation and Comparison of Text Compression Algorithms in Image Steganography”, *International Conference on Engineering Technologies (ICENTE'17)*, Konya, Türkiye, 2017, pp. 944-947. (Yüksek lisans tezinden yapılmıştır.)

Cataltas O., Tutuncu K., “Improvement of LSB Based Image Steganography”, *Proceedings of Research World International Conference*, Rome, Italy, 2017, pp. 36-40. (Yüksek lisans tezinden yapılmıştır.)

Cataltas O., Tutuncu K., “Comparison of LSB Image Steganography Technique in Different Color Spaces”, *International Artificial Intelligence and Data Processing Symposium*, Malatya, Türkiye, 2017, pp. 1-6. (Yüksek lisans tezinden yapılmıştır.)

Tutuncu, K., Cataltas, O., “Diagnosis of Mesothelioma Disease Using Different Classification Techniques”, *7th International Conference on Advanced Technology & Sciences (ICAT’17)*, İstanbul, Türkiye, 2017, pp. 346-350

Tutuncu, K., Koklu, M., Cataltas, O., “Comparison of classification techniques on leaf dataset”, *4th International Conference on Advanced Technology & Sciences (ICAT’Rome)*, Rome, Italy, 2016, pp. 144-148.

Tutuncu, K., Cataltas, O., Koklu, M., “Occupancy detection through light, temperature, humidity and CO2 sensors using ANN”, International Conference on Science, Technology, Engineering and Management, Rabat, Morocco, 2016, pp. 16-20.