Video Formatına Veri Gizleme Amacıyla Gömülmüş Bir Steganografi Uygulamasının Geliştirilmesi

Hasan BADEM[1] , Mahit GÜNEŞ[2]*

[1]Kahramanmaraş Sütçü İmam Üniversitesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi, Kahramanmaraş, Türkiye
[2]Kahramanmaraş Sütçü İmam Üniversitesi, Elektrik-Elektronik Mühendisliği, Kahramanmaraş, Türkiye

ÖZET: Gelişen teknoloji ile birlikte veri güvenliğinde yeni problemlerle karşılaşılmakta ve veri güvenliğini artırmak amacıyla, bu problemleri çözmeye yönelik girişimler yapılmaktadır.Yapılan araştırmalarda, veri güvenliğine yönelik problemler, matematiksel veya mantıksal önermelere dayanan veri şifreleme algoritmaları ile çözülmekte veya bir verinin farklı verilerin içerine mantıksal önermelere göre yerleştirilmesine dayanan steganografi algoritmaları geliştirilerek çözülmeye çalışılmaktadır. Bu çalışmalar sayesinde önemli veriler güvenlik altına alınabilmektedir. Steganografi tekniğine göre geliştirilen çoğu uygulamalar metin, ses ve resim üzerinde yoğunlaştığı görülmektedir. Bu çalışmada, gizlenmek istenen veri ilk olarak metin editöründen ASCII karakter formatında girilen ve gizlenmek istenen veri, ikili formata dönüştürülmektedir. İkinci aşamada, AVI formatındaki örtü videosunun kapasitesi oranında her frame'deki ilgili piksellerinin en önemsiz biti (LSB) kullanılarak, ikili formatındaki verinin ilgili bitiyle yerinin değiştirilmesi esasına dayanan bir steganografi uygulaması geliştirilmiştir.

Anahtar Kelimeler: *Steganografi, Video işleme, Veri gizleme, En önemsiz bit(LSB) yöntemi*

Developing of a Steganography Application for Data Hiding Embedded in Video Format

ABSTRACT: A variety of new problems are encountered as the result of developing technology for data security. In order to ensure security of data, new improvements are made to solve these problems. According to the researchers in the literature, data security problems are solved by mathematical or logical propositions, based on the data encryption algorithms or by improving stenographical algorithms, which functions as placing the data into different data according to logical propositions. Many important data can be retrieved by the studies on data security processes. Most applications developed according to Steganography technique have been concentrated on the text, sound and image. In this study, firstly, ASCII formatted and data to be hided is entered from text editor and then converted to binary format. In the second step, a stenography application is developed that depends on the changing the placement of the least significant bit (LSB) in each frame' in proportion of the cover video capacity in AVI format while converting the data to hide which is put in ASCII character format in text editor.

Keywords: *Steganography, Video processing, Data hiding, Least Significant Bit (LSB) method*

1. GİRİŞ

Gelişen teknoloji nedeniyle, verilerin korunması gün geçtikçe daha çok önem kazanmaktadır. Bu nedenle veri güvenliğini artırma üzerine son yıllarda önemli çalışmalar yapılmaktadır. Literatürde yer alan çalışmalar incelendiğinde, çeşitli algoritmaların geliştirildiği görülmektedir. Bu algoritmalardan Steganografi-tabanlı algoritmaların başarılı sonuçlar verdiği görülmektedir.

Steganografi, gizlenmek istenen bir verinin bir örtücü verinin içerisine saklanması tekniğine dayanan bir veri gizleme yöntemidir. Bu yöntemde gizlenen veri kod çözücü olmadığı sürece çözülebilmesi zordur. Bu yaklaşım, bir verinin bir nesne içerisine maskelenerek saklı tutulması veya gizlenmesi olarak da tanımlanmaktadır [4]. Steganografi, Dilbilim Steganografi ve Teknik Steganografi olmak üzere kendi içerisinde ikiye ayrılmaktadır. Dilbilim steganografi, taşıyıcı verinin metin (text) olduğu steganografi koludur. Teknik Steganografi ise birçok konuyu içine almaktadır. Bunlar; görünmez mürekkep, gizli yerler, microdot'lar ve bilgisayar tabanlı yöntemler gibi birçok konuyu içine almaktadır. Bilgisayar tabanlı yöntemler metin, ses ve resim dosyalarını kullanan veri gizleme yöntemleridir. Görüntü dosyaları içerisine saklanacak veriler metin dosyası olabileceği gibi, herhangi bir görüntü içerisine gizlenmiş başka bir görüntü dosyası da olabilir. Bu yaklaşımda içine bilgi gizlenen ortama örtü verisi(cover-data), oluşan ortama da stego-metin(stego-text) veya stego-nesnesi(stego-object) denilmektedir [2].

Bu çalışmada, video nesnelerinin içerisine gizlenmek istenen verinin steganografi algoritmasıyla saklanması gerçekleştirilmiştir. Bu sayede önemlilik derecesi yüksek olan bir verinin video içerisine

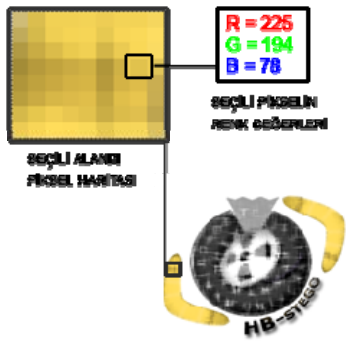
*Sorumlu Yazar: Mahit Güneş, mgunes@ksu.edu.tr

gömülerek gizlenmesi sağlanmıştır. Veriyi videodan çözecek program olmadıkça istenmeyen durumlara karşı veri koruma altına alınmıştır. Veri gömme işleminde bir videonun her bir frame'in her bir Pikselinin *Least Significant Bit* (LSB) yerine saklanmak istenen verinin ilgili bitinin eklenmesine dayanmaktadır.

1.1. Dijital Video

Bir video, frame adı verilen resimlerin saniyede ardışık şekilde gösterilmeleriyle oluşturulur. Bir videonun kalitesini belirleyen standart değerler vardır. Bunlar; saniyedeki işlenen frame sayısı (*frame per second-fps*-) ve frame'lerin çözünürlüğüdür ki kısaca frame'deki piksel sayısıdır. Örneğin, bir frame'in – resim- çözünürlüğü ve sayısal değeri Şekil 2.' de gösterilmektedir.

Bir resimdeki renkler, RGB olarak ifade edilen 3 temel rengin Kırmızı, Mavi, Yeşil karışımından oluşur. Bu renkler 8 bitlik sayısal değerler ve 256 adet ton alabilmektedir [5] . Bir resimdeki, rastgele seçilmiş bir pikselin görünümü ve sayısal değeri (RGB) Şekil 1' de gösterilmektedir. Bu resimde seçilen pikselin rengine ait sayısal değerler; kırmızı 225 , yeşil 194 ve mavi 78 değerliğe sahiptir.



Şekil 1. Bir Resimdeki Bir Pikselin Görünümü ve Sayısal Değeri

Şekil 2. Resmin Çözünürlüğü ve Sayısal Değeri

**Geliştirilen uygulama için tasarlanmış logo kullanılmıştır*

1.2. En Önemsiz Bit (Least Significant Bit -LSB)

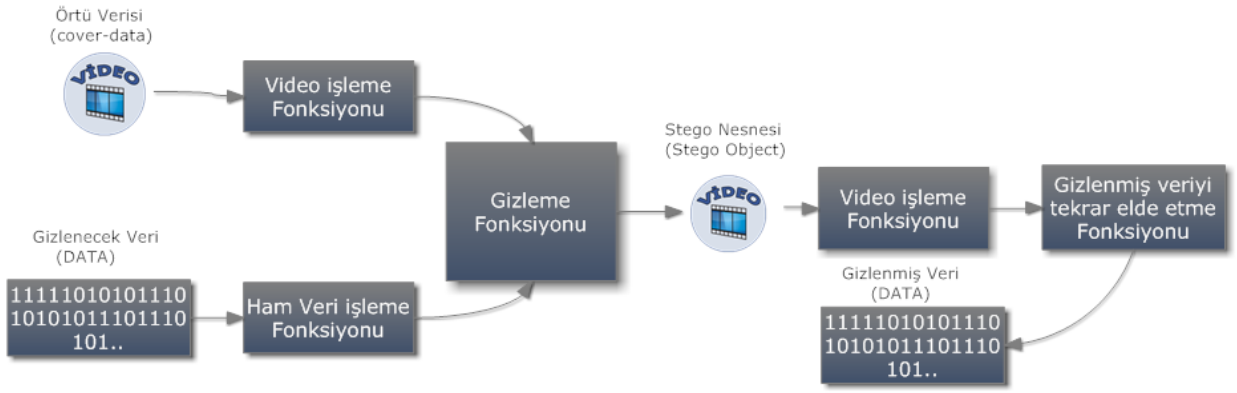
24 bitlik bir resminde bir piksel 3 byte'tan oluşmaktadır. Her pikselin rengi Kırmızı , Yeşil ve Mavi olmak üzere 3 ana renkten elde edilmektedir. Buna pikselin RGB değeri denmektedir. Her byte'ın son biti, ilgili rengin LSB'ine denk gelmektedir [13]. Şekil 1. de görünen rastgele seçilmiş pikselin RGB değeri Tablo 1. de gösterilmiştir. Bu çalışmada, üzerinde işlem yapılacak LSB biti mavi renk kanalı değerliğinin ilk bitine karşılık gelmektedir.

Tablo 1. LSB Bitinin Gösterilmesi

Renk	***Onlu***	***İkili***
Kırmızı	225	11100001 **LSB**
Yeşil	194	11000010 **LSB**
Mavi	78	01001110 **LSB**
RGB	111000011100001001001110 **LSB**	

2. MATERYAL VE METOT

Son zamanlarda yapılan veri gizleme (steganografi) ile ilgili araştırmalar incelendiğinde, bu çalışmaların hareketsiz resimler ve görüntüler üzerinde yoğunlaştığı görülmektedir [6-12]. Steganografi tekniği, gizleme işlemi, verinin saklanacağı taşıyıcı ortam ve gizlenecek veri olmak üzere iki parametreye sahiptir[3]. Bu çalışmada, bir video dosyasına verilerin gömülmesi prensibine dayalı bir steganografi algoritması geliştirilmiştir. Şekil 3'te blok diyagram halinde gösterilen algoritma iki ana kısımdan oluşmaktadır. Bu kısımlar sırasıyla veri gömme ve veriyi geri elde etmeden oluşmaktadır. Geliştirilen algoritma, *"kmaras.avi*" dosyası üzerinde test edilmiştir.



Şekil 3. Algoritmanın Genel Blok Şeması

2.1. Veri Gizleme

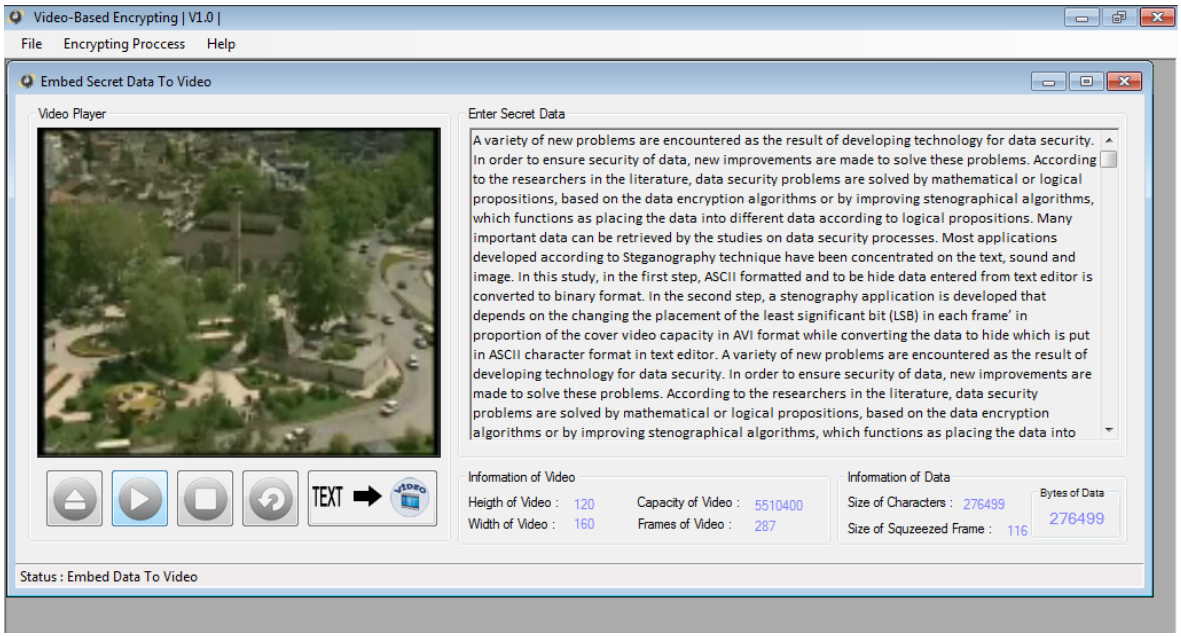
Geliştirilen algoritmanın veriyi gizleme kısmında, 3 temel işlemden oluşturulmuştur. Bunlar sırasıyla videoyu analiz eden ve frame'leri elde etme işlemi, metin editöründen girilen ASCII formatındaki veriyi ikili formata çevirme işlemi ve bu işlemlerden sonra, frame'lere ikili formatındaki veriyi gömen gizleme işlemleridir.

Video analizi işlemi; videonun frame sayısını, her frame 'in çözünürlüğünü ve bu iki parametreyle videoya toplam gizlenecek veri kapasitesinin hesaplama işlemlerini içermektedir. Video kapasitesi; denklem (1) üzerinden hesaplanmaktadır.

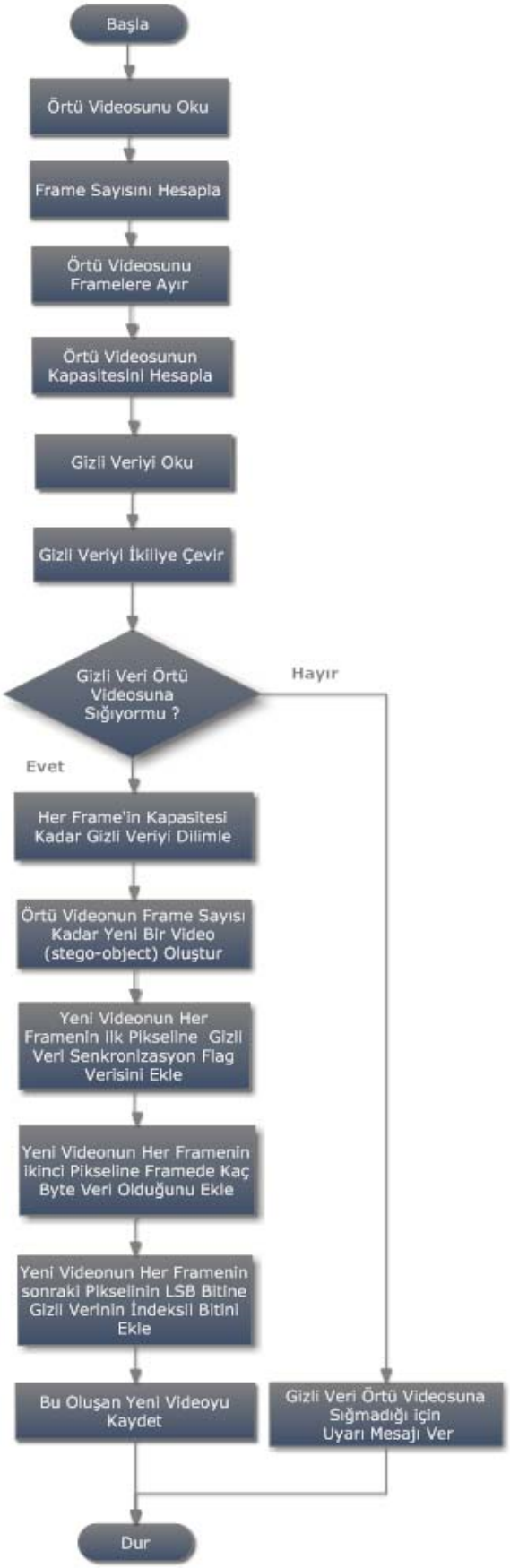
$$Video\ Veri\ Kapasitesi = \frac{Frame\ Sayısı\ *\ Bir\ Frame'in\ çözünürlüğü}{8} \quad (1)$$

Metin editöründen girilen ASCII karakter formatındaki veri, ikili sayısal değerliğe dönüştürülür. İkili değerlikteki veri ise, ilgili pikselin LSB'sine eklenerek videoya verileri saklamaktadır.

Bu işlemlerin sonunda geliştirilen örtüleme algoritmasına ait akış diyagramı Şekil 5'te gösterilmiştir. Bu algoritma kullanılarak elde edilen yeni görüntü için Şekil 4'te görülen bir ara yüz ekranı tasarlanmıştır. Ara yüz ekran görüntüleri alınırken, metin editörüne, bu makalenin *abstract* kısmının yinelenerek yazılması ile oluşturulan veri kullanılmıştır.



Şekil 4. Geliştirilen Uygulamanın "Gizli Datanın Bir Örtü Videosuna Gizlenme" Ekran Görüntüsü



Şekil 5. Gizli Datanın Bir Örtü Videosuna Gizlenme Algoritması Akış Diyagramı

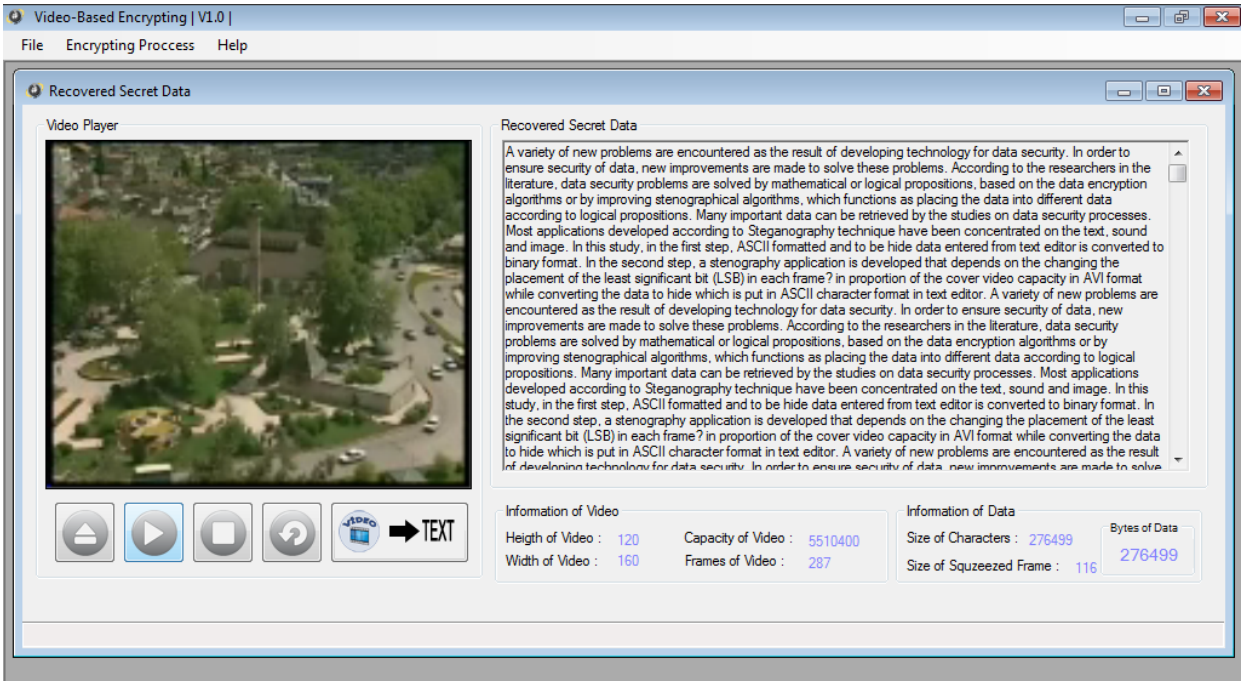
2.2. Veriyi Geri Elde Etme

Geliştirilen algoritmanın videodan veriyi geri elde etme kısmı, iki temel işlemden oluşmaktadır. Bunlar sırasıyla, verinin gizlendiği videoyu analiz etme ve bu videodaki gizlenmiş piksellerin LSB'lerinden elde edinilen ikili formattaki veriyi ASCII karakter formatına çevirerek gizlenmiş olan veriyi ortaya çıkartma işlemleridir.

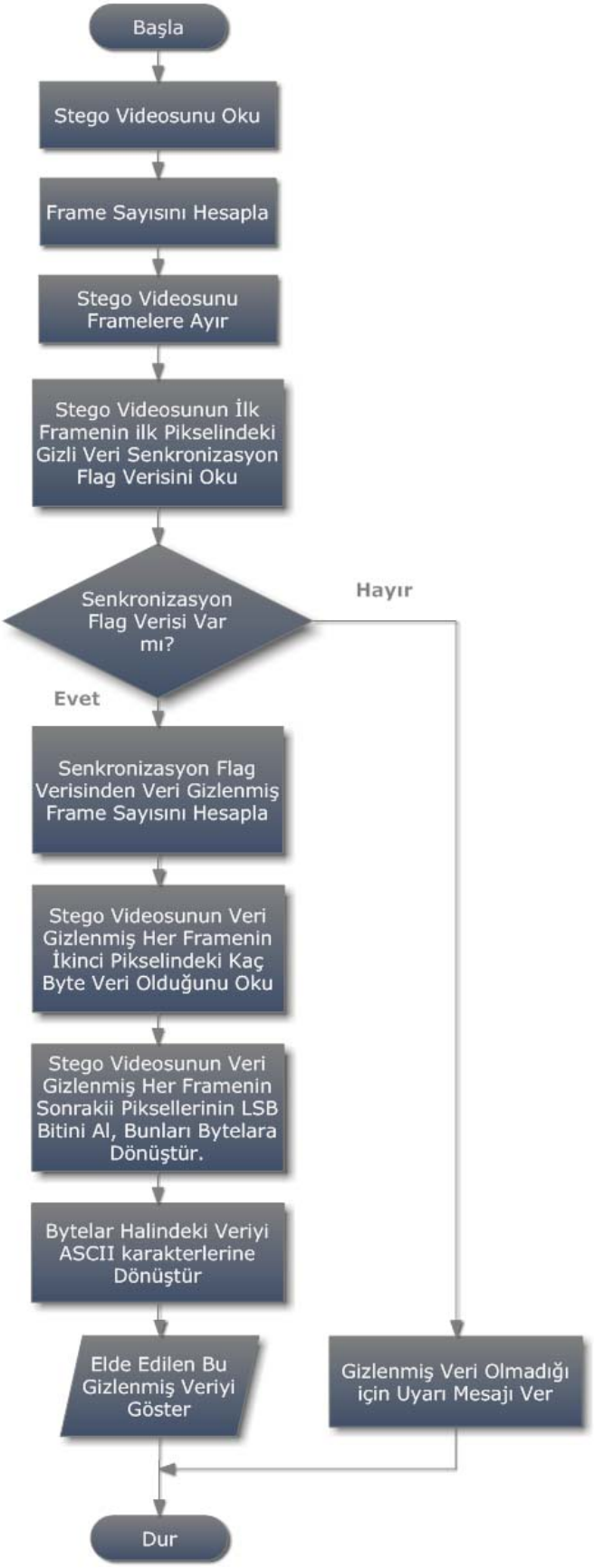
Videoyu analiz etme işleminde; videoya bu algoritmayla veri gizlenmişliğini belirleme, videonun veri gizlenmiş frame sayısını, her frame'in çözünürlüğünü, videonun içerdiği gizlenmiş veri kapasitesini hesaplama işlemlerini içermektedir.

Videodan gizlenmiş veriyi çıkartma işleminde; Videonun veri gizlenmiş piksellerinden elde edilen bit halindeki veriyi ASCII karakterlerine çevirerek gizlenmiş veriyi ortaya çıkarma işlemlerini içermektedir.

Bu işlemlerin sonunda geliştirilen çıkartım steganografi algoritmasının akış diyagramı Şekil 7'de gösterilmiştir. Bu algoritma kullanılarak geliştirilen sistem için Şekil 6'da görülen arayüz tasarlanmıştır. Arayüz ekran görüntüleri alınırken daha önceden videoya gizlenen ve metin editöründe bu makalenin *abstract* kısmının yinelenerek yazılması ile oluşturulan veri geri elde edilmiştir.



Şekil 6. Geliştirilen Uygulamanın "Gizli Datanın Video'dan Geri Elde Etme" Ekran Görüntüsü



Şekil 7. Bir Stego Videosuna Gizlenen Verinin Geri Elde Etme Algoritması Akış Diyagramı

2.3. Geliştirilen Sistemin Performans Analizi

Geliştirilen uygulama, *"kmaras.avi"* video dosyası kullanılarak analiz edilmiştir. Bu video, 287 adet frame ve 160x120 çözünürlüğüne sahip frame'lerden oluşmaktadır. Geliştirilen sistemin performans analizi için, bu makalenin *abstract* bölümünün yinelenmesi sonucu elde edilen veri kullanılmıştır. Bu veri 160x120 çözünürlüğünde yaklaşık 116 frame'e sığacak büyüklüktedir. Ayrıca bu değerler, program görüntüleri olan Şekil 4 ve Şekil 6'da gösterilmiştir. Geliştirilen sistemde; histogram analizi, verinin LSB üzerinden oluşturulma analizi ve sistemin cevap süresi analizleri olmak üzere üç farklı analiz yapılmıştır.

2.3.1. Histogram Analizi

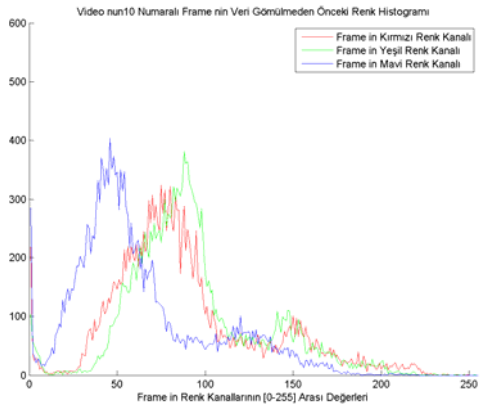
Steganografi uygulamaları, gizlenecek olan veri piksellerinin RGB değerlerinin değiştirilmesi, diğer bir ifadeyle, LSB değeri yerine verinin ilgili bitinin eklenmesi mantığına dayandığı için, örtü videosunun renk değerleri değişmektedir. Bu yüzden frame'lerin renk yoğunluğunun değişimi önem kazanmaktadır. Renk yoğunluğu farkı, denklem (2) ye göre hesaplanmaktadır. Bu denklemde, *"A"* veri gömülmeden önceki frame'i, *"B"* ise veri gömüldükten sonraki frame'i ifade etmektedir[1].

$$D(A,B)=\sum_{i=0}^{255}|H_A(i)-H_B(i)| \qquad (2)$$

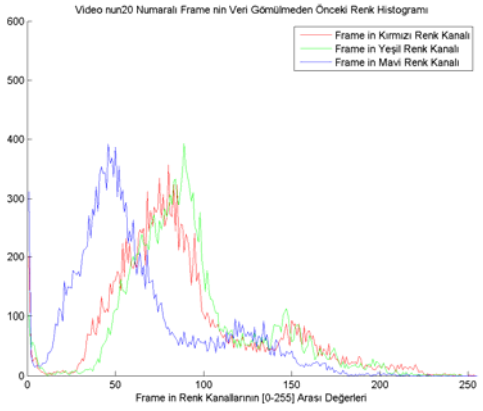
Renk yoğunluğu farkı analizi, test videosunun onuncu ve yirminci frame'leri örnekleme yapılarak gerçekleştirilmiştir. Örnekleme frame'leri, resim görünümleri, histogramları ve renk yoğunlukları, yani histogram farklarını gösteren grafikler Şekil 8'de gösterilmiştir.

(a1) Onuncu frame'in veri gizlenmeden önceki.görünümü

(b1) Yirminci frame'in veri gizlenmeden önceki.görünümü



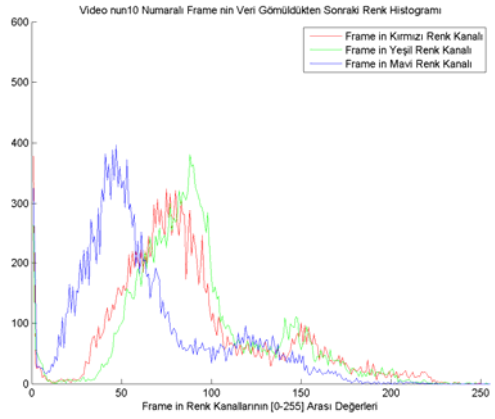
(a2) Onuncu frame'in veri gizlenmeden önceki histogramı



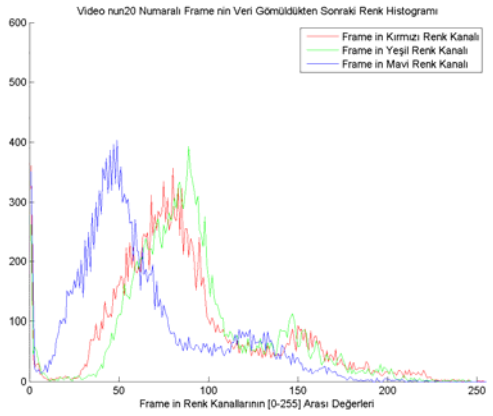
(b2) Yirminci frame'in veri gizlenmeden önceki histogramı

(a3) Onuncu frame'in veri gizlendikten sonraki. görünümü

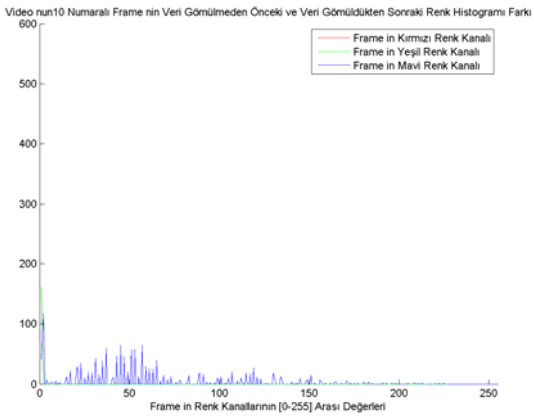
(b3) Yirminci frame'in veri gizlendikten sonraki. görünümü



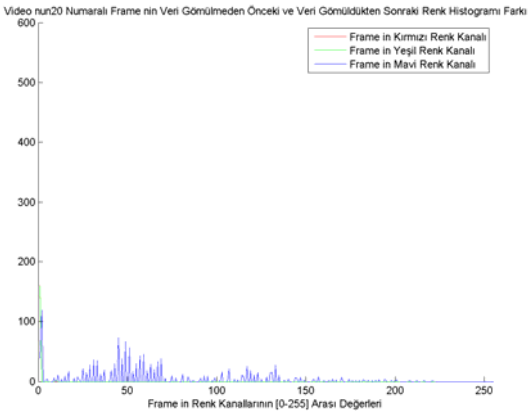
(a4) Onuncu frame'in veri gizlendikten sonraki Histogramı



(b4) Yirminci frame'in veri gizlendikten sonraki.histogramı



(a5)Onuncu frame 'in histogramlarının farkı



(b5) Yirminci frame 'in histogramlarının farkı

Şekil 8. Geliştirilen sistemde test videosunun onuncu ve yirminci Frame'lerinin veri gizlenmeden önceki ve gizlendikten sonraki görünümlerini ve histogram analizleri.

Şekil 8 incelendiğinde test videosunun örnekleme frame'lerinin işlem öncesi ve sonrası görünümlerinde gözle fark edilemeyecek kadar küçük değişimler olduğu görülmektedir. Ayrıca her videonun *Frame per second –fps-* değeri ortalama 24 olduğu için video izlenirken fark edilme ihtimalini neredeyse yok etmektedir. Histogram farkları incelendiğinde, kırmızı ve yeşil kanalların aynı olduğu, üzerinde işlem yapılmadığını ve piksellerin LSB'si yani mavi kanal üzerinde işlem yapıldığı görülmektedir. Bu da sistemin amacına hizmet ettiğini göstermektedir.

2.3.2. Verinin LSB Üzerinden Oluşturulma Analizi

Şekil 4'te gösterildiği gibi metin editöründen test amacıyla girilen veri, Şekil 6'da gösterildiği gibi geri elde edilmiştir. LSB üzerinden verilerin video gizlenmesi ve tekrar elde edilmesini analiz etme için test videosunun birinci frame'i örneklem olarak seçilmiştir. Bu frame'in işlem öncesi ve sonrası görünümleri Şekil 9'da ve LSB değişimleri örnek pikseller üzerinden Tablo 2.de gösterilmiştir.

(a)

(b)

Şekil 9. Birinci frame'in (a) işlem öncesi görünümü (b) işlem sonrası görünümü

Tablo 2. Birinci Frame'in işlem öncesi ve işlem sonrası [(0,0)-(0,7)] arası piksellerinin LSB Değişimi

	İşlem öncesi (0,0) (0,1) (0,2) (0,3) (0,4) (0,5) (0,6) (0,7)							**İşlem sonrası** (0,0) (0,1) (0,2) (0,3) (0,4) (0,5) (0,6) (0,7)						
	K		***Y***		***M***		***LSB***	***K***		***Y***		***M***		***LSB***
	On.	*İki.*	*On.*	*İki.*	*On.*	*İki.*		*On.*	*İki.*	*On.*	*İki.*	*On.*	*İki.*	
(0,0)	*0*	*0000*	*1*	*0001*	*0*	*0000*	***0***	*0*	*0000*	*1*	*0001*	*0*	*0000*	***0***
(0,1)	*0*	*0000*	*2*	*0010*	*1*	*0001*	***1***	*0*	*0000*	*2*	*0010*	*1*	*0001*	***1***
(0,2)	*0*	*0000*	*0*	*0000*	*0*	*0000*	***0***	*0*	*0000*	*0*	*0000*	*0*	*0000*	***0***
(0,3)	*0*	*0000*	*0*	*0000*	*0*	*0000*	***0***	*0*	*0000*	*0*	*0000*	*0*	*0000*	***0***
(0,4)	*8*	*1000*	*4*	*0100*	*5*	*1001*	***1***	*8*	*1000*	*4*	*0100*	*4*	*0100*	***0***
(0,5)	*6*	*0110*	*2*	*0010*	*3*	*0011*	***1***	*6*	*0110*	*2*	*0010*	*2*	*0010*	***0***
(0,6)	*0*	*0000*	*0*	*0000*	*0*	*0000*	***0***	*0*	*0000*	*0*	*0000*	*0*	*0000*	***0***
(0,7)	*1*	*0001*	*1*	*0001*	*1*	*0001*	***0***	*1*	*0001*	*1*	*0001*	*1*	*0001*	***1***

Tablo 2 incelendiğinde, birinci frame'deki işlem öncesi ve sonrasındaki ilgili sekiz pikselin değişimi görünmektedir. İşlem sonrasındaki ilgili piksellerin LSB değerlerinden ikili sayı sistemindeki ***"01000001"*** verisi elde edilmektedir. Bu değeri ondalık sayı sistemine dönüştürüldüğünde ***"65"*** sayısı elde edilir. Bu sayıda ASCII tablosunda ***"A"*** karakterine karşılık gelmektedir. Bu karakter Şekil 4 ve Şekil 6'da gösterildiği gibi test videomuza gizlenen verinin ilk karakterini oluşturmaktadır. Bu durum, sistemin LSB üzerinden veriyi işlemiş olduğunu göstermektedir.

2.3.3. Sistemin Cevap Süresi Açısından Analizi

Geliştirilen sistemin performansına etki eden unsurlardan birisi de işlem süresince geçen zamandır. Çünkü veriyi gizleme ve tekrardan geri elde etme işlemleri gerçek zamanlı olarak çalışmaktadır. Sistemde 5 farklı boyuttaki verilerin test videosuna gizlenerek, geri elde edilme işlemleri esnasında geçen süreler ve veriler ile ilgili bilgiler Tablo 3'te yer almaktadır.

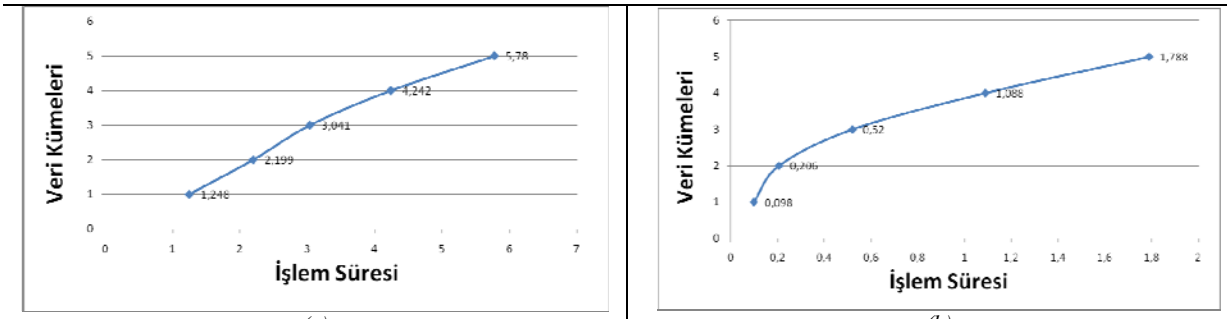
Tablo 3. Sistemin gerçek zamanlı cevap süresinin bazı veri kümeleri üzerinden analizi**

Veri Küme Numarası	**Videoya Gizlenen Veri ile İlgili Bilgiler***			**Videodan Çıkartılan Veri ile İlgili Bilgiler***		
	Karakter Sayısı	Yerleştiği Frame Sayısı	İşlem süresi (sn)	Karakter Sayısı	Yerleştiği Frame Sayısı	İşlem süresi (sn)
1	*2167*	*1*	***1,248***	*2167*	*1*	***0,098***
2	*4335*	*2*	***2,199***	*4335*	*2*	***0,206***
3	*10839*	*5*	***3,041***	*10839*	*5*	***0,520***
4	*21678*	*10*	***4,242***	*21678*	*10*	***1,088***
5	*32519*	*14*	***5,780***	*32519*	*14*	***1,788***

**Test videosu 287 frame ve 160*120 çözünürlüğe sahip olmak üzere toplam veri kapasitesi = 5510400 byte 'dır.*

***Bu analiz,3GB ram,Intel core 2DuoP8700 işlemcili(2.53 Ghz).ATI Radeon HD4570 (1.5 GB paylaşımlı belleği var) ekran kartı, 5400 rpm'lik 240 GB kapasiteli HDD' si olan ve Windows 7 işletim sistemi yüklü bir bilgisayarda üzerinde yapılmıştır.*

Tablo 3'te yer alan bilgiler üzerinden sistemin veri gizleme ve veriyi geri elde etme işlemlerindeki cevap süreleri ile veri kümelerinin boyutuna göre ilişkileri Şekil 10'daki grafiklerde görülmektedir.



(a) *(b)*

Şekil 10. Sistemin gerçek zamanlı cevap süresinin Tablo 3 de görülen veriler üzerinden (a) veri gizleme (b) veriyi geri elde etme işlemlerindeki cevap süreleri ile veri kümelerinin boyutlarına göre ilişkisi

Şekil 10 incelendiğinde sistemin veriyi videoya gizleme işlemi, veriyi geri elde etme işlemine oranla daha yavaş olduğu görülmektedir.

3. SONUÇ

Son yıllarda teknolojinin gelişmesiyle ortaya çıkan veri güvenliği sorununu çözmeye yönelik çeşitli çalışmalar geliştirildiği görülmektedir. Bu çalışmalarda steganografi çalışmaları öne çıkmaktadır. Çünkü steganografi, önemli olan verileri, başka veri kümelerinin içine gizleyerek verinin elde edinirliğini çok düşük seviyeye indirmektedir. Bu çalışmada, veri güvenliği sorununun çözümüne steganografi-tabanlı yeni algoritmalar geliştirilerek test edilmiştir. Bu çalışmada, gizlenecek olan veri olarak metin kullanılmıştır. Geliştirilen sistemin histogram analizi, verilerin gizlenme ve elde edilmesindeki video veri analizi ve sistemin cevap süreleri analizinde sistemin performans analizleri ortaya konulmaya çalışılmıştır. Bu analizlerin sonucunda, sistemin başarılı olarak amacına hizmet ettiği görülmektedir. Bu çalışma sonucunda, veri güvenliği sorununa Steganografi-tabanlı çalışmalar veri gizleme çözümleri getirilerek sonuç alınabileceği görülmüştür.

Steganografi yaklaşımlarını ister on-line ister off-line olarak hizmet veren web tabanlı sistemler üzerinde, uygulayarak veri transferlerinde verilerin güven altına alınabileceği düşünülmektedir.

REFERANSLAR

[1] Bender W, Gruhl D, Morimoto N, Lu.A *" Techniques for data hiding*". IBM Systems Journal, 35(3-4):313-36,1996.

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

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

[4] J.Fridrich, and M.Goljan, "*Practical steganalysis: state-of-the-art*," Proceeding of SPIE Photonics West, Electronic Imaging 2002, San Jose, California, vol.4675, pp.1-13, January 2002.

[5] Kakumanu P, Makrogiannis S, Bourbakis N. "*A survey of skin-color modeling and detection methods"*. Pattern Recognition: pp: 1106–22, 2007.

[6] Ozdemir C, Ozcerit A.T. "*A new steganography algorithm based on color histograms for data embedding into raw video streams*", Computers & Security Vol(28), 670-682, 2009.

[7] Artz, D.; , *"Digital steganography: hiding data within data,"* Internet Computing, IEEE , vol.5, no.3, pp.75-80, May/Jun 2001

[8] Noda, H.; Furuta, T.; Niimi, M.; Kawaguchi, E.; , *"Application of BPCS steganography to wavelet compressed video,"* Image Processing, 2004. ICIP '04. 2004 International Conference on , vol.4, no., pp. 2147- 2150 Vol. 4, 24-27 Oct. 2004

[9] Hanafy, A.A.; Salama, G.I.; Mohasseb, Y.Z.; *"A secure covert communication model based on video steganography,"* Military Communications Conference, 2008. MILCOM 2008. IEEE , vol., no., pp.1-6, 16-19 Nov. 2008

[10] Mozo, A.J.; Obien, M.E.; Rigor, C.J.; Rayel, D.F.; Chua, K.; Tangonan, G.; , *"Video steganography using Flash Video (FLV),"* Instrumentation and Measurement Technology Conference, 2009. I2MTC '09. IEEE , vol., no., pp.822-827, 5-7 May 2009

[11] Eltahir, M.E.; Kiah, L.M.; Zaidan, B.B.; Zaidan, A.A.; , *"High Rate Video Streaming Steganography,"* Information Management and Engineering, 2009. ICIME '09. International Conference on , vol., no., pp.550-553, 3-5 April 2009

[12] Changyong Xu; Xijian Ping; Tao Zhang; , *"Steganography in Compressed Video Stream,"* Innovative Computing, Information and Control, 2006. ICICIC '06. First International Conference on , vol.1, no., pp.269-272, Aug. 30 2006-Sept. 1 2006

[13] Oğuz C. *"Görüntü İşaretleri İçin Yeni Bir Sayısal Damgalama Yöntemi"* Doktora Tezi, İstanbul Üniversitesi, pp 15-16, 2006