İleri Teknoloji Bilimleri Dergisi
Journal of Advanced Technology Sciences
ISSN:2147-3455

STEGANOGRAFİ İÇİN EN UYGUN RESMİ BELİRLEYEN UYGULAMA ARAYÜZ TASARIMI

UMAS 2017'de sunulmuş ve genişletilmiş bildiridir.

Mehmet Zeki KONYAR, Sümeyya İLKİN, Nazlıcan ÇELİK, Adnan SONDAŞ

Kocaeli Üniversitesi, Teknoloji Fakültesi, Bilişim Sistemleri Mühendisliği Bölümü, 41380, Kocaeli, TÜRKİYE
mzeki.konyar@kocaeli.edu.tr

Özet- Steganografi (gizliyazı olarak da bilinir), gizli mesajları insan gözüyle fark edilmeyecek biçimde taşıyıcı dosyanın içerisine saklayarak gönderen bir gizli haberleşme yöntemidir. Steganografide amaç taşıyıcı dosyanın değişim miktarını çıplak gözle fark edilmeyecek kadar düşük tutmak ve en fazla miktarda gizli veriyi alıcıya yollamaktır. Bu çalışmada, önerilen yöntem ve tasarlanan arayüz yardımıyla gizli mesajın hangi resimde (taşıyıcıda) en az değişikliğe sebep olacağı tespit edilmektedir. En az oranda bit değişikliğine uğrayan resim tespit edilip kullanıcıya o resim önerilmektedir. Veri gizleme işlemi için üç boyutlu resmin renk kanallarının hepsi kullanılarak büyük bir kapasite sağlanmaktadır. Önerilen yöntemin en önemli özelliklerinden birisi de gizli veriyi geri elde ederken alıcının orijinal görüntüye ihtiyaç duymamasıdır. Dolayısıyla orijinal görüntü haberleşme kanalında veya alıcıda asla bulunmayacaktır. Ayrıca alıcıya gönderilecek görüntü gizlenecek mesaja göre farklılık gösterdiğinden, seçilecek olan resmin önceden bilinme ihtimali de bulunmamaktadır.

Anahtar Kelimeler- Gizli haberleşme, LSB, Steganografi, Çoklu resim.

APPLICATION INTERFACE DESIGN TO SELECT OPTIMAL PICTURE FOR STEGANOGRAPHY

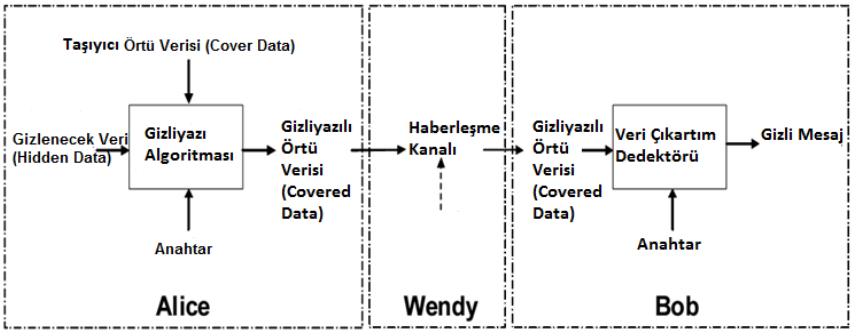
Abstract-Steganography is a secret communication method which sends hidden messages inside the carrier file unnoticeably with human eye. Steganography (also known as data hiding) aims to keep the amount of change of the carrier file low enough not to be noticed by the naked eye and to send the maximum amount of hidden data to the recipient. In this study, with the help of the proposed method and the designed interface it is determined that in which picture (carrier) the secret message will cause the least change. The picture which caused the least change is detected and it is suggested to the user. For data hiding operation a large capacity is provided with use of all color channel of the three-dimensional Picture. One of the most important features of the proposed method is that the receiver does not need the original image while the hidden data is recovered. Therefore, the original image will never be found in the communication channel or receiver. Also, since the image to be sent to the recipient differs according to the message to be hidden, there is no possibility to previously know of which image will be selected for data hiding.

Key Words- LSB, Multiple images, Secret communication, Steganography.

1. GİRİŞ (INTRODUCTION)

Veri gizleme ilk çağlardan beri gizli haberleşme için kullanılan bir yöntemdir. Günümüzde gizliyazı (steganografi) ve damgalama (watermarking) veri gizleme biliminin en çok kullanılan yöntemleridir. Steganografi daha çok gizli haberleşme amacıyla kullanılırken, damgalama genellikle ticari amaçlarla ve telif hakkı işlemleri için kullanılmaktadır. Veri gizlemede amaç istenmeyen kişilere gizli mesajın varlığını hissettirmemektir. Şekil 1'deki mahkûm problemi [1] kullanılarak steganografi yöntemi tasvir edilmektedir. Bu problemde mahkûmlar (Alice ve Bob), sıradan iletişim kuruyormuş gibi görünmektedirler. Fakat aslında steganografi kullanarak gardiyana (Wendy) hissettirmeden gizli haberleşme yapmaktadırlar.

Steganografi için taşıyıcı örtü verisi (cover data) olarak resim, ses ve video benzeri dosyalar kullanılmaktadır. Alıcı tarafından belirli bir gizliyazı algoritması kullanılarak ve duruma göre bir anahtar şifre ile bu örtü dosyası içerisine gizli veri saklanır. İçerisine veri gizlenmiş dosya, örtülü veri (covered data) veya gizliyazılı örtü verisi olarak adlandırılmaktadır. Haberleşme kanalı aracılığıyla alıcıya gönderilen örtülü veri burada veri çıkartım dedektörü olarak adlandırılan, gizliyazı algoritmasını çözen, bir yöntemle (kullanıldıysa anahtarı da devreye koyarak) gizli mesajı çıkartır ve gizli haberleşmeyi tamamlar [2-6].



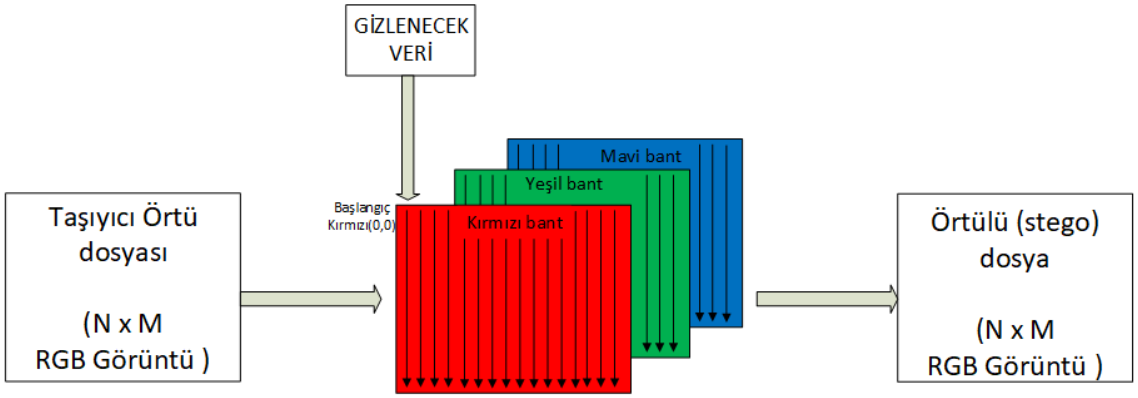
Şekil 1. Mahkûm problemi ve steganografi süreci (Prisoner problem and steganography process)

2. YÖNTEM (METHOD)

Önerilen çalışmada görüntülere veri gizlenirken yukarıdaki avantajları sebebiyle LSB yöntemi tercih edilmiştir. LSB yöntemi güvenilir olmakla beraber gizliyazı analizi (steganaliz) işlemi aşamasında bazı şüpheler doğurabilir [5]. Bundan dolayı önerilen çalışmada, görüntüdeki bozulmanın olabildiğince az olabilmesi için örtü dosyası birden fazla resim içerisinden en uygun olan belirlenmektedir.

2.1. Önerilen Veri Gizleme Yöntemi (Proposed Data Hiding Method)

LSB ile veri gizleme işleminde genelde bit ekleme işlemi için ilgili pikselin ikilik sistemdeki son bitine gizli bit yazılır. Şekil 2'de, önerilen veri gizleme yöntemi taşıyıcı örtü dosyası olarak üç boyutlu (RGB) görüntüleri kullanmaktadır.



Şekil 2. Önerilen veri gizleme yöntemi (Proposed data hiding method)

Gizlenecek bilgi, RGB görüntünün kırmızı bandının sütunlarına yukarıdan aşağı doğru yazılmaya başlanır. Her sütun bittiğinde sonraki sütuna geçilerek devam edilir. Gizli metnin uzunluğuna göre yeşil ve mavi bantlarda aynı şekilde kullanılabilir. Verinin elde edilmesi esnasında bilginin bittiğinin anlaşılabilmesi için gizlenecek verinin sonuna NULL karakteri eklenmektedir.

LSB işlemi için ilk olarak gizlenecek metin ele alınır. İlgili metnin her bir harfine karşılık gelen onluk sistem ASCII kodları belirlenir. Daha sonra bu ASCII kodların ikilik sistemde karşılıkları tespit edilir. Tablo 1'de "kocaeli" gizli kelimesi için bu anlatım örneklenmiştir. Son olarak bu ikilik sayı dizisinin uzunluğu belirlenip, kırmızı bandın ilk pikselinden başlayarak Şekil 2'de gösterildiği gibi sütunlar boyunca bu uzunluğa denk gelen piksellerin son bitleri (LSB) sıfırlanır.

Tablo 1. LSB veri gizleme örneği (LSB data hiding example)

	kocaeli	Gizli metin
Örnek:	107, 111, 99, 97, 101, 108, 105	Harflerin ASCII Kod karşılıkları (Onluk)
	01101011, 01101111, 01100011, 01100001, 01100101, 01101100, 01101001	Harflerin ASCII Kodları (İkilik)

Gizlenecek verinin ikilik kod karşılığının ilk biti kırmızı renk kanalının ilk *(0,0)* pikseline (1) ve (2)'de gösterilen şekilde gizlenir. Burada Δ son biti sıfırlanan piksele eklenecek bit değerini , "*m*" ise gizlenecek bit değerini göstermektedir. Bir sonraki bit için alt satırdaki piksele (1,0) geçilerek aynı işlem tekrarlanır. Sütun sonuna gelindiğinde bir sonraki sütuna geçilerek devam edilir. Bu işlem son bit gizlenene kadar tekrar eder.

$$piksel_{yeni} = piksel_{sonbit\_sıfır} + \Delta \quad (1)$$

$$\Delta = \begin{cases} 1 & ; eğer\ "m = 1" \\ 0 & ; eğer\ "m = 0" \end{cases} \quad (2)$$

Veri gizleme işlemi tamamlandıktan sonra alıcı tarafta gizlenen verinin doğru olarak elde edilebilmesi için, gizlenen verinin sonuna eklenen NULL karakterinin ikilik ASCII kod karşılığı da aynı mantıkla gizlenir. Böylece gizliyazı içeren örtülü dosya hazırlanmış olur.

2.1. Veri Çıkartımı (Data Extraction)

Alıcı tarafında örtülü dosya alındıktan sonra içerisinde bulunan gizli metnin çözülmesi işlemi veri gizleme algoritmasına bağlı olarak belirlenmektedir. Bu çalışmada alıcı gizli bitleri çıkarırken ilk

olarak veri gizleme işlemine benzer biçimde kırmızı renk kanalının ilk pikselinden başlanarak sütunlar boyunca her bir pikseldeki gizli biti (3)'teki gibi elde edilir.

$$gizlibit = \begin{cases} 1 & ;eğer\ "pikseldeğeri_{sonbit} = 1" \\ 0 & ;eğer\ "pikseldeğeri_{sonbit} = 0" \end{cases} \quad (3)$$

Bu işlem NULL karakterinin ikilik ASCII kodu elde edilene kadar devam eder. Sonra Tablo 1'deki örneğin tersten işletilmesi gibi; NULL karakterinden önceki ikilik bit dizisinin her bir sekiz bitinin onluk ASCII karşılıkları belirlenir ve gizlenen veri elde edilir. Tablo 2'de, sekiz piksel değeri kullanılarak "*k*" harfinin elde edilişi gösterilmiştir.

Tablo 2. LSB veri çıkartma örneği (LSB data extraction example)

Örnek:	*180* *181* *175* *176* *173* *180* *181* *181*	Örtülü görüntünün kırmızı bant birinci sütunundaki ilk 8 elemanı
	01101011	En üstten başlanarak her bir pikselin parite bitleri (soldan sağa yazılır)= ikilik ASCII kodu
	107	ASCII Kodu(onluk)
	k	Harf karşılığı

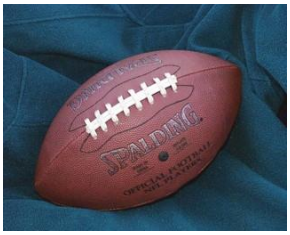
3. DENEYSEL ÇALIŞMALAR (EXPERIMENTAL STUDIES)

Bu çalışmada, gizlenmek istenen verinin en az miktarda bit değişimine neden olacağı taşıyıcı resim tespit edilmektedir. Veri gizlemeden kaynaklanan bozulmanın en az oranda olmasını sağlamak için bit değişimi en az olan örtülü dosya seçilerek alıcıya gönderilmektedir. Yapılan çalışmada, taşıyıcı görüntü olarak Matlab paket programının [7] çeşitli çözünürlüklere sahip olan Şekil 3'teki örnek görüntüleri kullanılmıştır.


1-Autumn (345×206)


2-Concordaerial (3060×2036)


3-Fabric (640×480)


4-Football (320×256)


5-Gantrycrane (400×264)

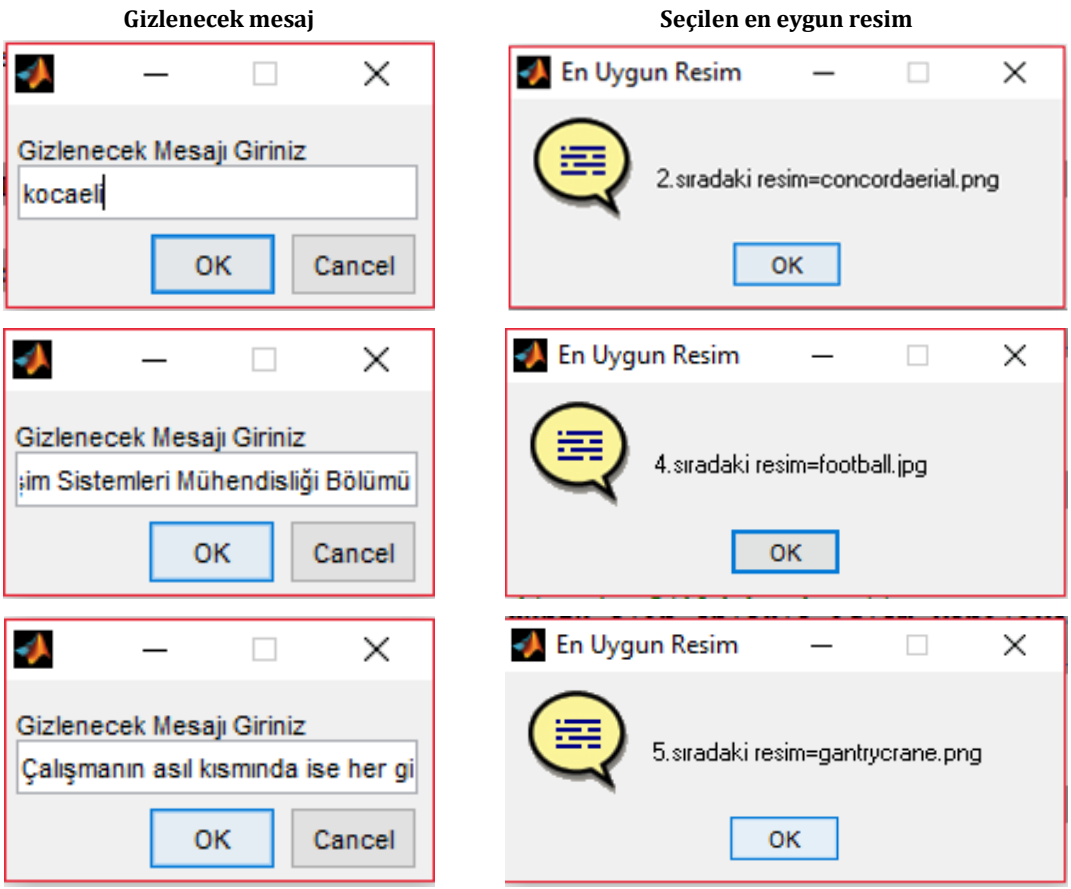

6-Greens (200×300)

Şekil 3. Veri gizleme için kullanılan örnek görüntüler (Sample images used for data hiding)

Veri gizleme işlemi önceki bölümde anlatılan yöntem kullanılarak yapılmıştır. Çalışmada öncelikle önerilen yöntemin doğruluğunu test için rastgele seçilen bir taşıyıcı dosyaya bazı veriler gizlenmiştir. Daha sonra alıcı tarafında bu gizli kelimelerin alınıp alınamayacağı kontrol edilmiştir. Tablo 2'de alıcının içerisine önerilen yöntemle "kocaeli" gizli kelimesi yazılan

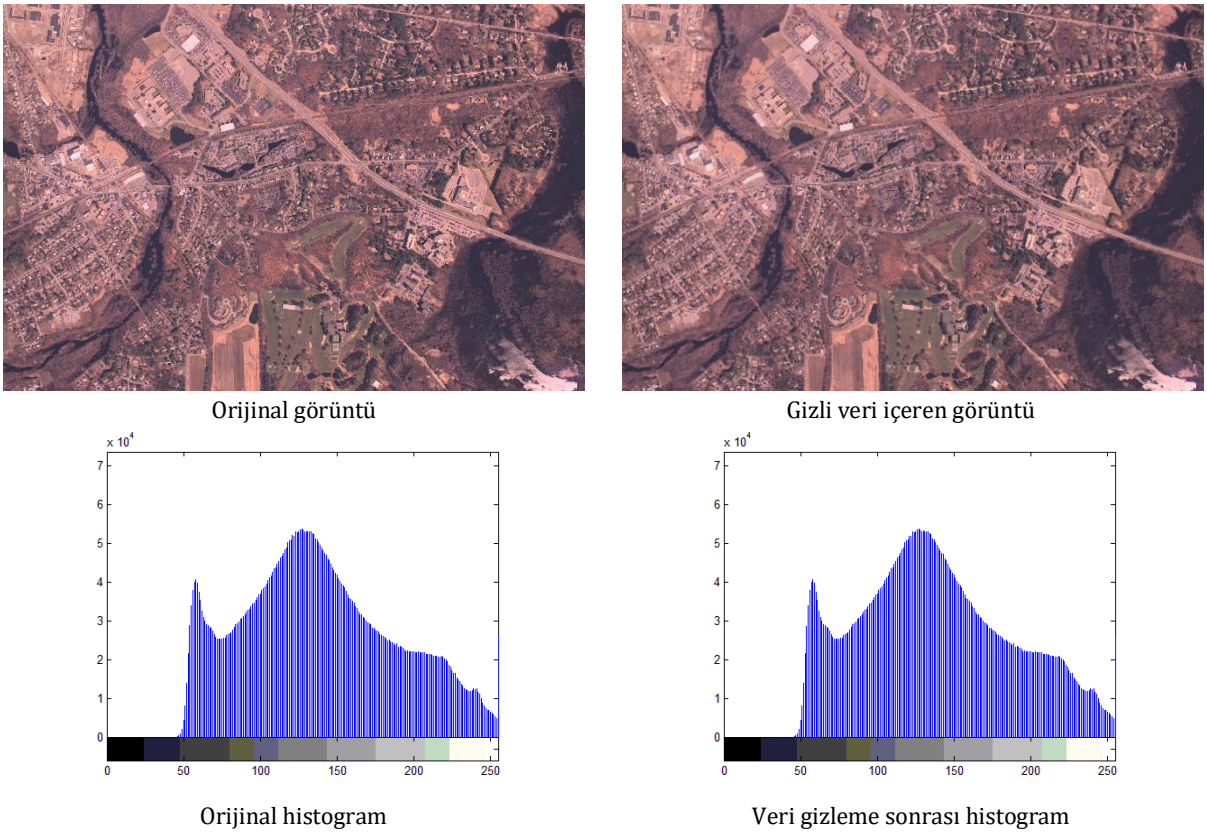
görüntüden veri çıkartımı örneklenmiştir. Buradan da anlaşılacağı üzere önerilen yöntem sayesinde kaynak görüntüye ihtiyaç duyulmadan gizli veri rahatlıkla elde edilebilmektedir.

Çalışmanın asıl kısmında ise aynı gizli veri Şekil 3'teki görüntülerin her birine sırayla gizlenerek bit değişim oranları kontrol edilmiştir. Geliştirilen arayüz sayesinde, veri gizleme sonrası en az değişime uğrayan görüntü belirlenmekte ve alıcıya bu görüntü kullanılarak gizli verinin iletilmesi önerilmektedir. Şekil 4'te sırasıyla "kocaeli", "Kocaeli Üniversitesi, Bilişim Sistemleri Mühendisliği Bölümü" ve yirmi kelimelik uzun bir cümlenin gizli veri olarak kullanılması durumunda uygulamanın seçtiği en uygun görüntü gösterilmektedir.



Şekil 4. Gizlenen veriye göre seçilen en uygun taşıyıcı görüntü (Optimal carrier image selected based on hidden data)

Şekil 5'te yaklaşık beş bin bite denk gelen bir gizli mesaj için, seçilen en uygun görüntü, o görüntünün orijinal hali ve histogram değerleri gösterilmiştir. Görüntülere ve histogram değerlerine dikkat edildiğinde orijinal görüntü ile taşıyıcı görüntü arasında görsel olarak bir fark olmadığı, histogramlar arasında ise fark edilmeyecek derecede küçük değişiklikler olduğu görülmektedir.



Şekil 5. . Görüntü ve histogram karşılaştırmaları (Image and histogram comparisons)

4. SONUÇ VE TARTIŞMA (CONCULUSION AND DISCUSSION)

Bu çalışmada taşıyıcı dosyanın değişim miktarını çıplak gözle fark edilmeyecek kadar düşük tutup, en fazla miktarda gizli veriyi hedefe yollamak amacıyla bir yöntem önerilmiştir. Birim zamanda gönderdiği veri miktarının fazla olması ve örtü verisinde oluşturduğu gürültü miktarının düşük olması sebebiyle LSB yöntemi tercih edilmiştir. Bu çalışmada, tasarlanan arayüz yardımıyla gizlenmek istenen verinin hangi resimde (taşıyıcıda) en az değişiklik meydana getireceği tespit edilmektedir. En az oranda bit değişikliğine uğrayan resim tespit edilip hedefe o resmin gönderilmektedir. Veri gizleme işlemi için üç boyutlu resmin renk kanallarının hepsi kullanılarak büyük bir kapasite sağlanmaktadır. Önerilen yöntemin en önemli özelliklerinden birisi de alıcının orijinal görüntüye ihtiyaç duymadan gizli veriyi rahatlıkla geri elde edebilmesidir. Dolayısıyla orijinal görüntü hiçbir zaman haberleşme kanalında veya hedef alıcıda bulunmayacaktır. Ayrıca hedefe gidecek olan görüntü gizlenecek mesaja göre farklılık gösterdiğinden veri gizleme için seçilecek resmin önceden bilinme şansı da bulunmamaktadır. Böylece herhangi bir güvenlik zafiyeti veya mesajın istenmeyen kişilerin eline geçme imkânı oluşmamaktadır.

Deneysel çalışmaların sonuçları incelendiğinde veri gizlenmiş görüntüde bozulma olmadığından istenmeyen kişiler tarafından görüntüyle ilgili herhangi bir şüphe de oluşmamaktadır. Bu çalışmanın devamında veri gizleme kapasitesinin ve görüntü kalite metriklerinin iyileştirilmesi planlanmaktadır.

5. KAYNAKLAR (REFERENCES)

[1]. Simmons, G. J., (1984). The Prisoners' Problem and the Subliminal Channel, *Advances in Cyrptology*, 51-67.

[2]. Li, B., He, J., Huang, J. and Shi, Y. Q., (2011). A Survey on Image Steganography and Steganalysis, *Journal of Information Hiding and Multimedia Signal Processing*, 2(2), 142-172.

[3]. Hussain, M. and Hussain, M., (2013). A Survey of Image Steganography Techniques, *International Journal of Advanced Science and Technology*, 54, 113-124.

[4]. Hamid, N., Yahya, A., Ahmad, R. B., and Al-Qershi, A. M., (2012). Image Steganography Techniques: An Overview, *International Journal of Computer Science and Security*, 6(3), 168-187.

[5]. Ertürk, İ., Yalman, Y., Çetin, Ö. and Akar, F., (2014). *Veri Gizleme*, Beta, İstanbul.

[6]. Sadek, M. M., Khalifa, A. S. and Mostafa M. G. M, (2015). Video steganography: a comprehensive review, *Multimedia Tools and Applications*, 74(17), 7063-7094.

[7]. The MathWorks Inc., (2013), *MATLAB Release 2013b*, Natick, Massachusetts, United States.