*http://fbe.trakya.edu.tr/tujs*
*Trakya Univ J Sci, 7(1): 17-22, 2006*
*ISSN 1305-6468*
*DIC: 183ASET710605060606*

**Araştırma Makalesi / Research Article**

# 24-BİT RENKLİ RESİMLER ÜZERİNDE EN ÖNEMSİZ BİTE EKLEME YÖNTEMİNİ KULLANARAK BİLGİ GİZLEME

Andaç ŞAHİN, Ercan BULUŞ, M.Tolga SAKALLI
Trakya Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, EDİRNE
Tel: 02842353985, e-mail: andacs@trakya.edu.tr

Alınış: 2 Mayıs 2005
Kabul Ediliş: 25 Kasım 2005

**Özet:** Günümüzde gelişen teknoloji ile birlikte dijital ortamdaki (metin, ses ve görüntü dosyaları) verilerin korunma ihtiyacı ortaya çıkmaktadır. Verilerin korunmasında ve gizlenmesinde şifreleme ve steganografi önemli rol oynamaktadırlar. Şifreleme mesajın içeriğinin korunması ile ilgilenirken steganografi mesajın varlığının gizlenmesi ile ilgilenmektedir. Dolayısıyla steganografi bir şifreleme yöntemi değil şifrelemeyi tamamlayıcı bir öğedir. Bu çalışmada BMP (Windows Bitmap) formatındaki 24-bit renkli resim dosyaları üzerinde en önemsiz bite ekleme yöntemi kullanılarak geliştirilen bir steganografi uygulaması anlatılmaktadır.

***Anahtar Kelimeler:*** Bilgi gizleme, LSB yöntemi, Steganografi

## Information Hiding Using LSB Insertion Method on 24-Bit Colored Images

**Abstract :** Nowadays, because of developing technology, the necessity of protecting data in digital media (text, audio and image files) has occurred. As a consequence, encryption and steganographic techniques take the responsibility on protecting and hiding the data. While cryptography is about protecting the content of messages, steganography is about concealing their existence. Steganography is not an encryption technique. Nevertheless, it is a complementary technique and can be used with encryption techniques. In this study, a steganography application on 24-bit colored image in BMP (Windows Bitmap) format using LSB (Least Significant Bit) insertion method has been described.

***Key words:*** Information Hiding, LSB methods, Steganography

## Giriş

Steganografi bilgi gizleme yöntemlerinin önemli bir alt dalıdır (Petitcolas, Anderson ve Kuhn, 1999). Bu yaklaşım, bir nesnenin içerisine bir verinin gizlenmesi olarak tanımlanabilir. Bu yaklaşımla ses, sayısal resim, video görüntüleri üzerine veri saklanabilir. Görüntü dosyaları içerisine saklanacak veriler metin dosyası olabileceği gibi, herhangi bir görüntü içerisine gizlenmiş başka bir görüntü dosyası da olabilir. Bu yaklaşımda içine bilgi gizlenen ortama örtü verisi (cover-data), oluşan ortama da stego-metin (stego-text) veya stego-nesnesi (stego-object) denmektedir.

Steganografi kelimesi Yunanca "steganos: gizli, saklı" ve "grafi: çizim yada yazım" kelimelerinden gelmektedir. Steganografi, Antik yunan ve Heredot zamanına kadar uzanan oldukça eski bir veri gizleme yöntemidir. Heredot, İran Savaşları sırasında, kafasını kazıtıp kafa derisinin üzerine, gizli bir mesajın dövmesinin yapılmasına izin veren bir ulaktan bahsetmektedir. Mesaj yazıldıktan sonra ulak saçının uzamasını beklemekte, daha sonra ulak mesajı bekleyen kişiye ulaşmakta, kafasını tekrar tıraş etmekte, böylelikle mesaj ortaya çıkmaktadır. Bu yöntem bilinen ilk steganografi uygulamasıdır. Daha sonraki zamanlarda steganografi, harflere müzik notalarının atanması, I. ve II. Dünya Savaşlarında kullanılan mors kodları, II. Dünya savaşı esnasında başarıyla uygulanan görünmez mürekkeplerin kullanımı gibi uygulamalarla karşımıza çıkmaktadır (Katzenbeisser ve Petitcolas, 2000).

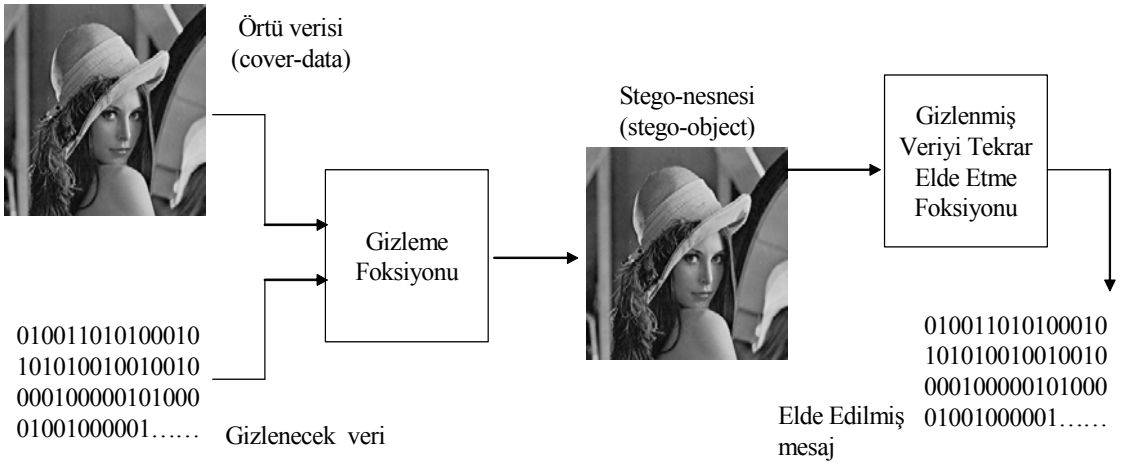
Günümüzde ise sayısal (dijital) nesneler üzerinde steganografi uygulamaları yapılmaktadır ve gelişen teknoloji nedeniyle, verilerimizi korumak amacıyla son yıllarda sıklıkla kullanılmaya başlanmıştır. Steganografi, Dilbilim Steganografi ve Teknik Steganografi olmak üzere kendi içerisinde ikiye ayrılmaktadır. Dilbilim steganografi, taşıyıcı verinin metin (text) olduğu steganografi koludur. Teknik Steganografi ise bir çok konuyu içine almaktadır. Bunlar; görünmez mürekkep, gizli yerler, microdot'lar, ve bilgisayar tabanlı yöntemler gibi başlıklar altında toplanabilmektedir. Bilgisayar tabanlı yöntemler metin, ses, görüntü, resim dosyalarını kullanarak veri gizleme yöntemleridir.

Steganografi kullanım alanları açısından üçe ayrılmaktadır. Bunlar aşağıdaki gibidir:

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

- Ses (audio) steganografi.

Görüntü dosyaları için bir steganografik sistem şekil 1'de gösterilmektedir. Gönderici bir gizleme fonksiyonu kullanarak bir steganogram yaratır. Gizleme fonksiyonu, verinin saklanacağı taşıyıcı ortam ve gizlenecek veri olmak üzere iki parametreye sahiptir (Westfeld ve Pfitzmann, 1999).




**Şekil 1.** Steganografik sistem

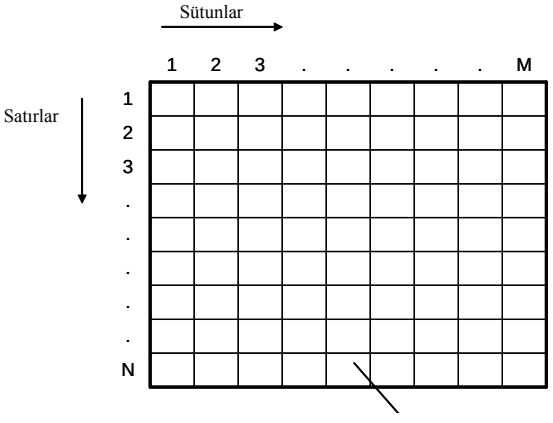
Görüntü steganografide, bilgilerin görüntü dosyaları içerisine saklanmasının çeşitli yöntemleri vardır. Şekil 1'de Gizleme Fonksiyonu olarak adlandırılan ve bilgi gizlemede en çok kullanılan yöntemler aşağıda gösterilmiştir:

- En önemsiz bite ekleme
- Maskeleme ve filtreleme
- Algoritmalar ve dönüşümler (Sellars, 1999).

Bu çalışmada en önemsiz bite ekleme - LSB yöntemi incelenmiş, BMP formatında 24 bit renkli resimler üzerinde çalışan Visual Studio .net'te bir uygulama geliştirilmiş ve çeşitli BMP dosyalar üzerinde ölçümler yapılmıştır.

**Sayısal Resmin Yapısı**

Sayısal (dijital) resim N satır ve M sütunluk bir dizi ile temsil edilir. Genellikle satır ve sütun indeksleri y ve x veya r ve c olarak gösterilebilir. Bir resim dizisinin elemanlarına piksel denir. En basit durumda pikseller 0 veya 1 değerini alırlar. Bu piksellerden oluşan resimlere ikili (binary) resim denir.




**Şekil 2.** Sayısal resmin temel yapısı

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

**En Önemsiz Bite Ekleme (LSB) Yöntemi**

En önemsiz bite ekleme yöntemi (Least Significant Bit Insertion Methods) yaygın olarak kullanılan ve uygulaması basit bir yöntemdir. Fakat yöntemin dikkatsizce uygulanması durumunda veri kayıpları ortaya çıkmaktadır. Bu yöntemde; resmi oluşturan her pikselin her byte'nın en önemsiz biti olan son biti değiştirilerek o bitin yerine gizlenmesini istediğimiz verinin bitleri sırasıyla verinin başlangıcından itibaren birer birer yerleştirilmektedir. Burada her sekiz bitin en fazla bir biti değişikliğe uğratıldığından ve eğer değişiklik olmuşsa da değişiklik yapılan bitin byte'ın en az anlamlı biti olmasından dolayı, ortaya çıkan steganogramdaki (= örtü verisi + gömülü veri) değişimler insan tarafından algılanamaz boyutta olmaktadır. Son bite ekleme işlemi resmin başından ya da sonundan olmak üzere sıralı bir

şekilde olabileceği gibi, bir rasgele fonksiyon üreteci (random function generator) kullanılarak belirlenen bir piksel üzerinde değişiklik yapılması şeklinde gerçekleştirilebilmektedir.

Bazı steganografik sistemler bazı gizli anahtarlar da kullanabilmektedir. Bu anahtarlar ikiye ayrılırlar:
1. *Steganografik anahtarlar;* mesajı resmin içine gizleme ve tekrar elde etme işlemini kontrol etme için kullanılırlar.
2. *Kriptografik anahtarlar;* Mesajın resmin içine gizlenmeden önce şifrelenmesi ve daha sonra deşifrelenmesinde kullanılırlar (Westfeld ve Pfitzmann, 1999).

**Gri-seviye Resimler ve LSB yönteminin uygulanması**

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

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

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

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

**8-bit Renkli Resimler ve LSB yönteminin uygulanması**

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

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

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

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

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

**24-bit Renkli Resimler ve LSB yönteminin uygulanması:**

24 bit resimler bir piksel başına 3 byte kullanmaktadır. Her pikselin rengi “Kırmızı (red), Yeşil (green), Mavi (blue)” olmak üzere üç ana renkten elde edilmektedir. Buna pikselin RGB değeri denmektedir. Her byte’ta son biti değiştirmek suretiyle bir piksel’de 3 bitlik bilgi saklanabilir. Yani 24 bit derinliğine sahip 1024x768 piksel boyutundaki bir resim, bilgi saklamak için kullanılabilir 2.359.296 bit (294.912 byte)’e sahiptir. Gizlenmek istenen mesaj, saklama işleminden önce sıkıştırılırsa çok daha fazla sayıda bilgi resmin içine gizlenebilir.

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

Orijinal görüntü bitleri yukarıdaki gibi verilen 3 pikselin içine “101101101” bilgisi gizlendiğinde oluşan yeni piksel değerleri aşağıdaki gibi olmaktadır.

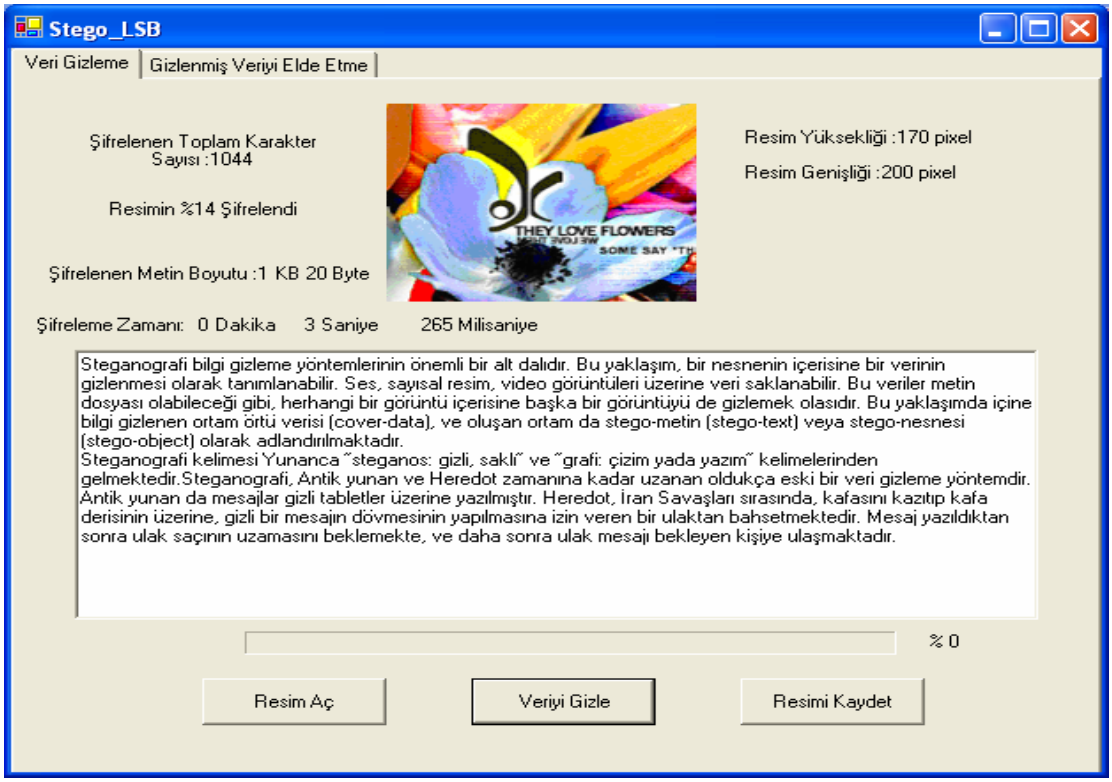
10010101 0000110**0** 11001001 (149,12,201)
1001011**1** 0000111**0** 1100101**1** (151,14,203)
10011111 00010000 11001011 (159,16,234)

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

**Uygulama**

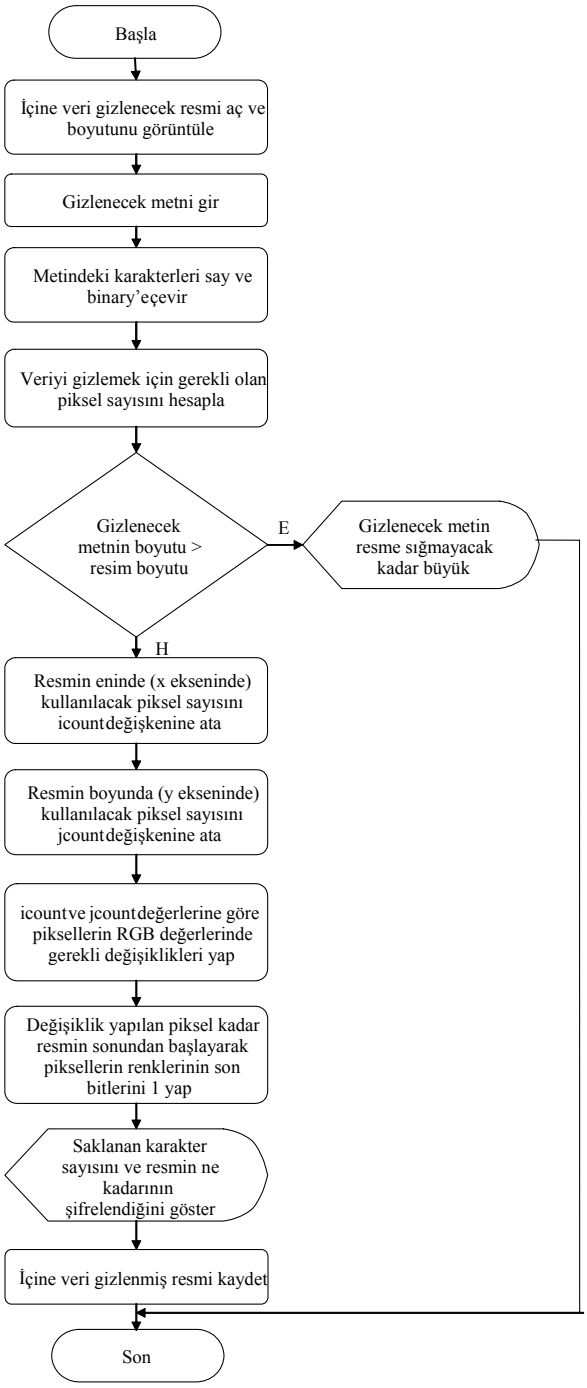
Bu çalışmada 24-bit renkli resimler üzerinde LSB yöntemini kullanarak verileri gizleyen program Visual Studio.net kullanılarak geliştirilmiştir. Programın *Veri Gizleme* kısmında öncelikle içine veri gizlenecek resim seçilmekte, daha sonra gizlenmesini istenen metin girilmekte ve şifreleme işlemi yapılmaktadır.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

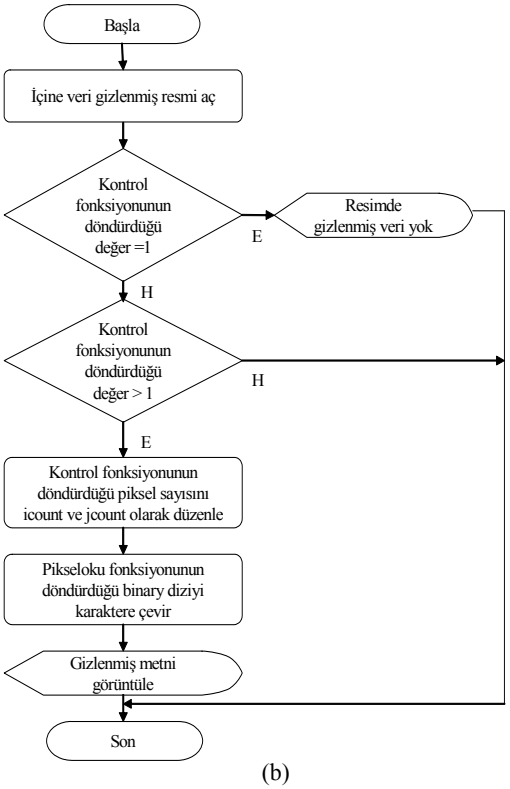
Programın örnek bir çalışma penceresi şekil 3'te gösterilmektedir. Burada 1044 byte boyutundaki bir metin resmin içine gizlenmektedir.

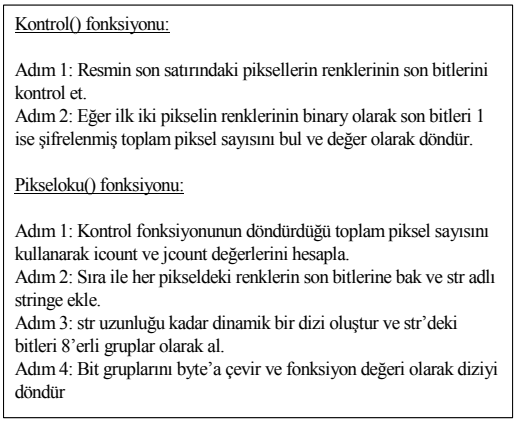



**Şekil 3.** Programın çalışma penceresi

Şekil 4'te, geliştirilen programın veri gizleme-gizlenmiş veriyi elde etme işlemlerinin akış şemaları ve programın kullandığı fonksiyonlar gösterilmektedir.




**Şekil 4.** (a) Programın Veri Gizleme işleminin temel akış şeması (b) Programın Gizlenmiş Veriyi Elde Etme işleminin temel akış şeması (c) Veriyi tekrar elde etme işleminde kullanılan fonksiyonların algoritması

Uygulamada önce 200x170 piksel - 99,6 KB boyutunda renkli bir resim kullanılmaktadır (resim1.bmp). Şekil 5, içine 1 KB (KiloByte) ve 5 KB büyüklüğünde metin gizleyerek elde edilen resimleri göstermektedir. Şekil 4 (a)'da gösterilen algoritma veri gizleme işlemi esnasında resmin kendisini de kullanmaktadır. Bu da gizlenen veri miktarının azalmasına yol açmaktadır. Dolayısıyla resim1.bmp'nin içine boyutundan dolayı en fazla 6,3 KB bilgi saklanabilmektedir. Daha büyük boyuttaki resimlerin içerisine gizlenen veri miktarı doğrusal bir artış göstermektedir.


(a) (b) (c)

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

Diğer bir örnek olan resim2.bmp ise 130x110 piksel - 41,89 KB boyutundadır. Bu resmin içine de aynı büyüklükte metinler gizlenmeye çalışılmıştır, fakat resmin boyutunun küçük olmasından dolayı 5 KB büyüklüğündeki metin içine gizlenememektedir. Resim2.bmp'nin içine saklanabilecek en fazla veri miktarı 2,7 KB'tır. Resim2.bmp'nin içine 1 KB ve 2,7 KB bilgi gizlenerek elde edilen sonuçlar da şekil 6'da gösterilmektedir.


(a) (b) (c)

**Şekil 6.** (a) Orijinal resim (130x110 piksel) (b) 1 KB saklanmış resim (c) 2,7 KB saklanmış resim

**Sonuç**

Bu çalışmada son bite ekleme yöntemini kullanan ve resmin içine metin verisi gizleyen bir program geliştirilmiştir. Program ile değişik boyutlardaki resimler kullanılarak değişik büyüklükte veriler resmin içine gizlenmiştir. Resmin boyutları arttırıldığında gizlenen verilerin büyüklüğünün de doğrusal olarak arttığı ve orijinal resim ile içine veri gizlenmiş resim arasında gözle görülür bir değişim olmadığı da gözlenmiştir. LSB yönteminin özelliği nedeniyle de steganografi uygulanan resmin boyutunda bir değişiklik olmamaktadır. Bu yaklaşımın dezavantajı gönderilecek mesajın veya dokümanın uzunluğunun resim boyutuna bağlı olmasıdır. Veri sıkıştırma işlemi yapıldıktan sonra gizleme işleminin gerçekleştirilmesiyle gizlenecek verinin miktarının arttırılması mümkün olacaktır. Saklanan verinin AES (Advanced Encryption Standard), RSA (Rivest- Shamir- Adleman) yada DES (Data Encryption Standard) gibi şifreleme yöntemleriyle şifrelenmesi sayesinde iletişimin daha güvenli yapılabilmesi de sağlanabilecektir.

**Kaynaklar**

1. FARID H., "Steganography: the art and mathematics of hiding information", Teaching Notes, Summer 2003. http://www.cs.dartmouth.edu/~farid/teaching/cs4/notes/steg.pdf
2. JOHNSON N. F., JAJODIA S., "Exploring Steganography: Seeing the Unseen", February 1998. http://www.jjtc.com/pub/r2026.pdf
3. KATZENBEISSER S., PETITCOLAS F.A.P., "Information Hiding Techniques for Steganography and Digital Watermarking", Artech House, INC. 685 Canton Street Norwood, MA 02062, 2000.
4. KESSLER G.C., "Steganography: Hiding Data Within Data", September 2001. http://www.garykessler.net/library/steganography.html
5. 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.
6. SAĞIROĞLU Ş., TUNÇKANAT M., "Güvenli İnternet Haberleşmesi İçin Bir Yazılım: Türksteg", 2002. http://mf.erciyes.edu.tr/turksteg/
7. SELLARS D., "An Introduction to Steganography", Student Papers, 1999. http://www.cs.uct.ac.za/courses/CS400W/NIS04/papers99/dsellars/index.html
8. 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.