Journal of the Faculty of Engineering and Architecture of Gazi University 38:3 (2023) 1725-1735

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University
Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

DLSB - The distanced least significant bit steganography

Burak Baysan[1], Serhat Özekes[2]*
[1]Department of Forensic Sciences, Institute of Dependence and Forensic Sciences, Uskudar University, 34662, Istanbul, Türkiye
[2]Department of Computer Engineering, Engineering Faculty, Uskudar University, 34662, Istanbul, Türkiye

Highlights:

- Pixel attack resistance
- Proportional distance bit placement
- Easy application

Keywords:

- Steganography
- lsb
- dlsb
- secret

Article Info:

Research Article
Received: 17.01.2022
Accepted: 09.08.2022

DOI:

10.17341/gazimmfd.1058824

Correspondence:

Author: Serhat Özekes
e-mail: serhat.ozekes@uskudar.edu.tr
phone: +90 216 400 2222

Graphical/Tabular Abstract

As a result of Pixel Attack on the image obtained after embedding the message with the LSB method with Figure A., the traces of the message placed at the bottom of the image are clearly seen. On the other hand, after the same message is placed on the same image with the DLSB method, it is clearly seen that there is no visible trace on the image, stacked in a certain area.

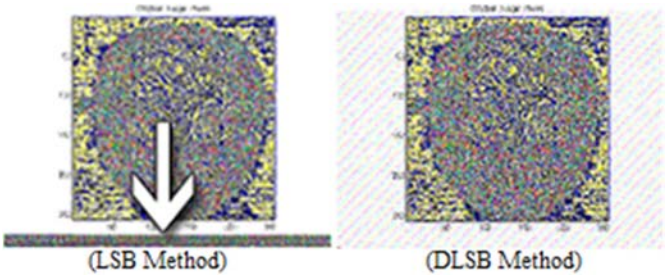


Figure A. LSB and DLSB Methods- %5 byte message – Pixel Attack

Purpose:

Today, LSB (*Least Significant Bit*) method is still used in many corporate, commercial and personal steganography tools that hide messages in 24-bit, 8-bit color and gray-scale images. The reason why this method is often preferred is that it can be applied very easily and it allows a very large message storage capacity. However, Pixel Attack, which can be easily applied against the LSB method and visualizes the traces of the embedded message, maintains its place in the literature and in the steganalist's toolbox. In this study, instead of continuous message bit placement used in LSB method, DLSB (*Distanced Least Significant Bit*) – Removed Most Significant Bit Steganography, which is a basic method that uses proportional distance message bit placement and is resistant to Pixel Attack, is proposed.

Theory and Methods:

With DLSB, it is used to place the message in the image with a proportional distribution / spacing, depending on the size of the message and the image. When embedding a $n_{bits}^{Message} = 8\left(n_{bytes}^{Message}\right)$-bit message into an n_{free}^{Image} byte image, each message bit is placed in the placement between the image bytes $Distance = \left(n_{free}^{Image} - \left(n_{free}^{Image} \bmod n_{bits}^{Message}\right)\right)/n_{bits}^{Message}$ is left blank. This $Distance$ value means that the message bits are proportionally spaced in the image pixels rather than stacked in a specific section. Here, the last bit of each $n^{Capacity}$ bytes at positions $54 + n^{Capacity}, 55 + n^{Capacity} + 1, \ldots, 54 + n^{Capacity} + n_{bits}^{Message}$ of the picture is replaced with the binary equivalent of the $Distance$ value. This is different from the LSB method, in which the distance between each message bit and the previous / next message bit is placed in the picture instead of the message size.

Conclusion:

Steganography performed on Bitmap image files with LSB method cannot resist Pixel Attack. Other steganography methods, which rely on compressing the message with various compression methods, are also affected by this attack. The DLSB method presented in this study applies a simple proportional distribution of message bits into the image. In the Pixel Attack, it was observed that the message was not piled up in a certain area in the image. DLSB is the first method to protect against Pixel Attack against LSB method. In addition, when DSLB and a compression method are used together, it has been observed that successful results are obtained in Peak Signal-to-Noise Ratio, R.S., Chi-Square, Sample of Pairs and Primary Sets analyzes.

Journal of the Faculty of Engineering and Architecture of Gazi University 38:3 (2023) 1725-1735

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

DLSB - Uzaklaştırılmış en önemsiz bit steganografi

Burak Baysan[1], Serhat Özekes[2]*

[1]Üsküdar Üniversitesi, Bağımlılık ve Adli Bilimler Enstitüsü, Adli Bilimler Anabilim Dalı, 34662, İstanbul, Türkiye
[2]Üsküdar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 34662, İstanbul, Türkiye

ÖNEÇIKANLAR

- Piksel saldırı direnci
- Orantılı uzaklıkta bit yerleşimi
- Kolay uygulama

Makale Bilgileri
Araştırma Makalesi
Geliş: 17.01.2022
Kabul: 09.08.2022

DOI:
10.17341/gazimmfd.1058824

Anahtar Kelimeler:
Steganografi
lsb
dlsb
gizli

ÖZ

Bugün halen 24-bit, 8-bit renkli ve gri-ölçekli imgeler içerisinde mesaj gizleyen çok sayıda kurumsal, ticari ve kişisel steganografi aracında LSB (*Least Significant Bit*) yöntemi kullanılmaktadır. Bu yöntemin sıklıkla tercih edilmesinin gerekçesi çok kolay uygulanabilmesi ve çok büyük mesaj saklama kapasitesine imkân vermesidir. Ancak LSB yöntemine karşı kolayca uygulanabilen ve gömülü mesajın izlerini görselleştiren Piksel Saldırısı ise literatürdeki ve steganalistin alet çantasındaki yerini korumaktadır. Bu çalışmada LSB yönteminde kullanılan aralıksız mesaj biti yerleşimi yerine, orantılı uzaklıkta mesaj biti yerleşimini kullanan ve Piksel Saldırısına dirençli temel bir yöntem olan DLSB (*Distanced Least Significant Bit*) – Uzaklaştırılmış En Önemsiz Bit Steganografi önerilmiştir.

DLSB - The distanced least significant bit steganography

HIGHLIGHTS

- Pixel attack resistance
- Proportional distance bit placement
- Easy application

Article Info
Research Article
Received: 17.01.2022
Accepted: 09.08.2022

DOI:
10.17341/gazimmfd.1058824

Keywords:
Steganography
lsb
dlsb
secret

ABSTRACT

Today, LSB (*Least Significant Bit*) method is still used in many corporate, commercial and personal steganography tools that hide messages in 24-bit, 8-bit color and gray-scale images. The reason why this method is often preferred is that it can be applied very easily and it allows a very large message storage capacity. However, Pixel Attack, which can be easily applied against the LSB method and visualizes the traces of the embedded message, maintains its place in the literature and in the steganalist's toolbox. In this study, instead of continuous message bit placement used in LSB method, DLSB (*Distanced Least Significant Bit*) Steganography, which is a basic method that uses proportional distance message bit placement and is resistant to Pixel Attack, is proposed.

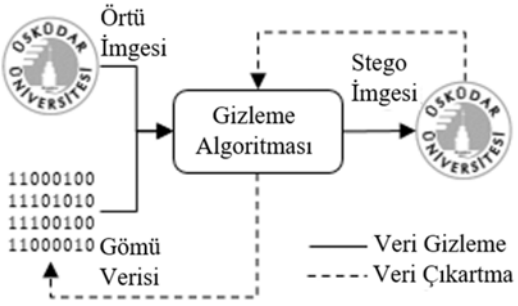
*Sorumlu Yazar/Yazarlar / Corresponding Author/Authors : burak.baysan@st.uskudar.edu.tr, *serhat.ozekes@uskudar.edu.tr / Tel: +90 216 400 2222

1. Giriş (Introduction)

İletişim güvenliği hem bireysel hem de devlet çalışmaları gibi kurumsal bilgilerin güvenliğinin sağlanması için önemlidir. Bu güvenliğin sağlanması için kullanılan ve hakkında çok sayıda çalışma da bulunan yöntemlerden birisi de steganografidir. Steganografi kelimesi Yunanca "*steganos: gizli*" ve "*graphy: yazı*" kelimelerinden gelmektedir. Steganografi bir nesnenin içerisine bir mesajın gizlenmesi olarak tanımlanabilir. Bu yöntemde imge, video, ses içerisinde mesaj saklanabilmektedir. Burada mesaja gömü verisi, içerisine mesaj yerleştirilen ortama örtü nesnesi, oluşan ortama da stego nesnesi denilir. Gömü verisi, örtü nesnesinin içerisine gizlenir ve gizlenen mesajı barındıran örtü nesne orijinalinden ayırt edilemez. Bu sayede de iki taraf arasındaki veri iletişiminin gizliliği ve güvenliği sağlanmaktadır. Veri örtme yönteminde üçüncü bir kişinin haberleşen iki kişiyi tespit etmesi oldukça zordur. İmge içerisine veri gizleme tekniklerinden en çok kullanılan ve en basiti olan en önemsiz bite veri gizleme tekniğidir ve bu yöntem birçok çalışmada tekrar çalışılmıştır. Bu teknik LSB (*Least Significant Bit - En Önemsiz Bit*) olarak adlandırılır [1]. LSB yönteminin güvenlik açısından çok sayıda zafiyeti vardır. Örneğin örtü verisine gürültü eklenmesi ile gömü verisi yok edilebilmektedir [1]. Veri gizleme yöntemlerinin analizini, gizlenen verinin ortaya çıkarılmasını ve tespitini sağlayan yöntemlere de steganaliz denilmektedir. Veri gizleme teknikleri iyi amaçlarla kullanılabildiği gibi kötü niyetli kişiler tarafından da kullanılabilmekte ve sonuçları büyük zararlara yol açabilmektedir. Steganaliz, çalışmaları kötü niyetli kullanımların önüne geçilmesi için örtü nesnesindeki gömü verisini tespit edebilmeyi amaçlar. Gömme işlemi sırasında çoğu durumda örtü nesnesinin boyunda bir değişiklik olmasa da veri yapısında değişiklikler oluşturulması gerekir. Steganaliz yöntemleri de veri gömme işleminin bıraktığı istatistiksel ve görsel izleri incelemektedir.

Steganografi yöntemleri Şekil 1 ile gösterildiği gibi kendi içerisinde sınıflandırılmaktadır. Bu sınıflandırmaya göre gömü verisi, örtü nesnesine gizleme yöntemiyle gizlenerek stego nesne oluşturulur. Stego nesne, örtü yani orijinal nesnenin formatını birebir ve duyusal özelliklerini aslına yakın olarak barındırır. Örtü nesnesinde yapılan değişimler insani algılarla fark edilemez düzeyde gerçekleşir. Stego nesne alıcıya iletildikten sonra, alıcı çıkartma yöntemini kullanarak gömü verisini elde eder.

İmge ortamına / örtü nesnesine, gömü verisinin gizlenmesiyle stego nesnesinin ve stego nesnesinden gömü verisinin elde edilmesini temsil eden akış Şekil 2 ile gösterilmiştir. Algoritma olarak bit uzayı sınıfına ve ortam / örtü nesnesi olarak imge sınıflandırmasına dahil olan ve basit olarak uygulanabilmesinden dolayı en yaygın kullanıma sahip olan yöntem LSB yöntemidir. LSB ile örtü imgesi ve gömü verisi ikili sayı sistemine çevrilerek gizleme işlemi gerçekleştirilir. Örtü imgesindeki her bir baytlık verinin en önemsiz biti değiştirildiğinden dolayı stego imge üzerinde insan gözü ile algılanabilecek bir değişiklik gerçekleşmez.



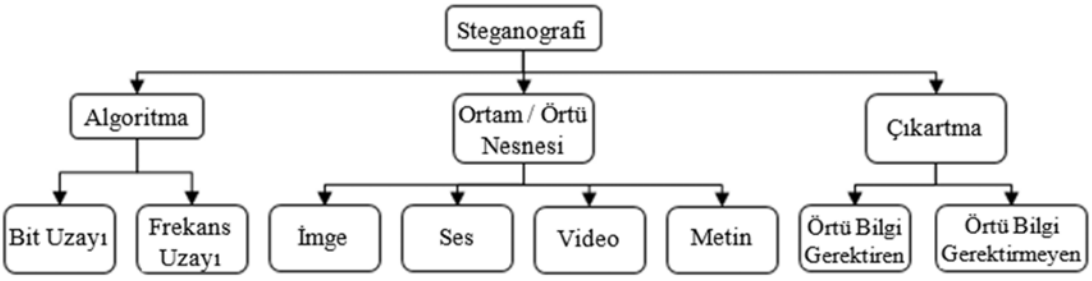
Şekil 2 İmge steganografi işlemi (Image Steganography Process)

Bu çalışmada önerilen DLSB – Uzaklaştırılmış En Önemsiz Bit Steganografi (*Distanced Least Significant Bit*) algoritma olarak bit uzayı, örtü nesnesi olarak imgeyi ve çıkartma olarak örtü nesnesi bilgisi gerektirmeyen sınıflarına dahil bir yöntemdir. DLSB Piksel Saldırısına doğrudan direnç sağlayan ve LSB yöntemini kullanan steganografi yöntemlerinde kullanılabilecek, LSB yönteminin bit yerleşim düzeninde geliştirilme yapılmış halidir.

Çalışma devamında bölüm 1.2 ile ilgili literatür, bölüm 2 ile LSB ve DLSB yöntemleri, bölüm 3 ile analizler ve son olarak bölüm 4 ile sonuçlar verilmiştir.

1.1. Literatür (Literature)

Steganografinin bilinen en detaylı tarihsel gelişim süreci 1967 yılında Kahn [2] tarafından yayınlanan çalışmayla açıklanmıştır. Daha spesifik bir tanım ise 2009 yılında Fridrich tarafından '*...gizlenmiş mesajın tespitini önleyen bilgi gizlenmiş objelerin sanatı ve bilimi...*' olarak verilmiştir [3]. Bu çalışmada önerilen DLSB bit uzayını kullanan LSB yönteminin geliştirilmiş durumudur, bu nedenle sınıfı temsil eden önemli literatür çalışmalarının incelenmesi önemlidir. Bit uzayını ve LSB kullanan yöntemlerden birisi olan eşleştirme yöntemi 2001 yılında Sharp [4] tarafından önerilmiştir. Sharp'ın önerdiği bu yöntemde ilkel-rastlantısal dizilerin oluşturulması için anahtar kullanan bir şema tasarlanmış ve anahtar dizi kullanarak gizli içerik ayrıca şifrelenerek gizlenmiştir. 2006 yılında Mielikainen [5] Sharp'ın [4] yöntemini geliştirmiş ve gizleme işlemini fonksiyondan alınan sonuca göre gerçekleştirmiştir. Bu yöntemde örtü imgesi piksel çiftlilerine ayrılmış ve gömü verisi ikişer bitlik gruplar halinde gizlenmiştir. Gömülecek ilk bit ilk pikselin son bitine doğrudan yerleştirilmiş, ikinci bit ise piksel çiftlisinin son bitlerinin gizleme fonksiyonundan aldığı sonuca göre yerleştirilmiştir. Bu fonksiyonun çalışması için piksel çiftlilerinden birisi 1 arttırılmış veya azaltılmıştır. Bu yöntem ile LSB yöntemine göre örtü imgesinde daha az değişiklik yapılırken, aynı kapasitede gömü verisi gizlenebilmektedir. Chan [6], Mielikainen'in [5] çalışmasını



Şekil 1 Steganografi (Steganography)

geliştirerek imgedeki ardıl iki pikselden ilk pikselin LSB biti ile ikinci pikselin LSB bitinden bir önceki bitine XOR yöntemini uygulamıştır. Tian [7] örtü imgede düşük bozulum, yüksek kapasite sağlayan ve tersinir veri gizleme yöntemini önermiştir. Bu yöntemde yüksek kapasite elde edilebilmiş ve örtü imgesine yüksek oranda veri gizlenebilmiştir. Tersinir olması nedeniyle de orijinal örtü imgesine ihtiyaç duymadan stego imgeden gömü veri geri elde edilebilmektedir. Tian'ın [7], önerdiği yöntemde örtü imgenin iki pikseli arasındaki fark iki kat genişletilerek oluşturulan alana gömü verisi gizlenmektedir. Alattar [8], Tian'ın yöntemi üzerinde geliştirme yaparak dört piksel arasındaki farkı iki kat daha genişletmiş ve 3-bitlik gömü verisini oluşan bu alana gizlemiştir. Chang vd. [9] çalışmalarında örtü imgesini iki kez oluşturarak, iki imgeye modül matrisi ve değişim yönünü kullanarak gömü verisini gizlediler. Yöntemde iki imge kullanılması sayesinde yüksek veri kapasitesi elde etmeyi başardılar. Lu vd. [10] Chang vd. [9] önerdiği yöntemi geliştirerek, örtü imgesinin pikselleri ile yüksek görsel kalite ve yüksek veri gömme kapasitesi elde ettiler. Ker [11] yaptığı çalışmada $2/3$ oranında verim sağlayan gizleme yöntemini önermiştir. Bu yöntemde iki bit veri gizleyebilmek için üç piksel kullanılmıştır. İki pikselin son biti veri gizleme için, son piksel gizlenen verinin aynı şekilde mi yoksa tümleyen olarak mı gizlendiğini göstermek için kullanılmıştır. Gizlenecek iki bit, örtü imgesinin son bitlerinde daha az değişiklik yapacaksa tümleyen olarak, değilse değişiklik yapmadan gizlenmektedir. Bu sayede de son bitlerde daha az değişiklik yapılarak veri gizlenmesi amaçlanmaktadır. Wu vd. [12] piksel farkını kullanarak taşıyıcı imge üzerindeki ardışık piksellerin çakışmayacak şekilde üst-üste getirilmesi ve piksel değerlerinin farklarının hesaplanmasıyla gerçekleşen veri gizleme yöntemini önerdiler. Burada ortaya çıkacak fark değerlerini farklı sınıflarla temsil ettiler. Bu sayede fark değerlerinin yerine yeni bir veri gizlenmesini sağladılar. Wang vd. [13] Wu vd. [12] çalışmaları üzerinde geliştirme gerçekleştirdiler. Bu yeni geliştirmede iki piksel arasındaki farkın modül fonksiyonu sonucu kullanılarak veri gizleme gerçekleştirildi. Fridrich ve Soukal [14] yaptıkları çalışmada kodlama teorisini temel alan ve hamming matrisini kullanan veri gizleme yöntemi önerdiler. Kurtuldu ve Arıca [15] yaptıkları ve imge kareleri yöntemi adını verdikleri çalışmalarında örtü imgesinin bloklara bölünerek gizlenecek verinin en yakın bit dizilimine sahip piksel grubuna yerleştirilmesi esasına dayalı çalışmalarını önerdiler, ancak bu yöntemde örtü imgesine gizlenebilen verinin kapasitesi oldukça azdır. Wu vd. [16] 2014 yılında gerçekleştirdikleri çalışmalarında stego imgenin iki kez oluşturulmasıyla ilk stego imgeye veriyi, ikinci stego imgeye gizlenen verinin meta verilerini gizleyen bir yöntemi önerdiler. Huang vd. [17] 2014 yılındaki çalışmalarında bitişik piksel çiftlisi eşleştirme yöntemini ve bitişik piksel çiftlisi seçimi için yeni bir rasgele seçim yöntemi uyguladılar. Sabeti vd. [18] çalışmalarında veri gizleme işleminde örtü verisi üzerinde güvenli bölgeyi belirleyebilmek amacıyla karmaşıklık ölçütü kullandılar ve bu yöntemin istatistiksel saldırılara karşı geleneksel yöntemlerden daha güvenli olduğunu gösterdiler. Jain ve Kumar [19] çalışmalarında gömü verisinin kayıpsız veri sıkıştırmasını ile sıkıştırılması ve devamında örtü imgesi içerisine yerleştirilmesini sağlayan yüksek kapasiteli veri gizleme yöntemini önerdiler. Atıcı ve Sağıroğlu [20] steganografi ile Windows işletim sistemi ortamında seçilen bir klasör ve içerisindeki dosyaları kilitlemeyi amaçlayan çalışmalarını duyurdular.

Literatürdeki steganaliz yöntemleri iki grup altında kategorize etmek mümkündür, ilki teknik-özgün steganaliz yöntemleri olan gömme algoritmasına özgün saldırılar [21] ve ikincisi evrensel steganaliz yöntemleri olan herhangi bir gömme algoritmasını hedef alan saldırılardır [22-25]. Diğer taraftan birçok steganografi yöntemine temel teşkil eden LSB yerleşim yöntemi ayrıca Piksel Saldırısına karşı büyük bir zafiyet içermektedir [26-29].

1.2. Motivasyon (Motivation)

Yukarıda verili literatür incelendiğinde imgeye veri gömme yönteminde LSB yöntemini temel alan çalışmalar yapıldığı görülmektedir. LSB'ye ek olarak rastlantısal piksel seçimine dayalı yöntemlerse rastlantısallık kabiliyetlerini çeşitli kriptosistemlerden ve/ya kaotik fonksiyonlardan almaktadır. Tekil olarak LSB'nin gömü verisini ardışık olarak örtü verisine yerleştirmeye karşı direnç sağlamayan bu yöntemler bir görsel saldırı yöntemi olan piksel saldırısına da direnç sağlayamamaktadır. LSB yönteminin hem kolay uygulanabilirliği hem de yüksek kapasitesi araştırmacıların ilgisini çekmeye devam ederken güvenlik tarafında imtiyazlar verilmesine neden olmaktadır. Bu çalışmada LSB'nin kolay uygulama ve kapasite avantajları korunarak güvenli hale getirilmesini amaçlayan DLSB yöntemi çalışılmıştır. DLSB temel bir yerleşim düzeni olarak verilmiş ve sonraki araştırmalara kaynak oluşturması amaçlanmıştır. Bu amacın korunması için çalışmaya herhangi bir kriptosistem ve/ya sıkıştırma yöntemi dahil edilmemiştir.

2. Yöntem (Method)

Dijital imgeler N satır ve M sütunluk bir dizi olarak temsil edilir. Bu dizi elemanlarına piksel denir. İmge dosyaları renkli olarak genellikle 8-bit ya da 24-bit, gri seviye imgeler 1-2-4-6 ya da 8-bit olabilmektedirler. Bu çalışmada da 8-bitlik üç bayt değerinden oluşan, her bir bayt değerinin sırasıyla kırmızı, yeşil ve mavi (*RGB – "Red Green Blue"*) renklerinin 0-255 arası tonlarını temsil ettiği 24-bit renkli BMP formatı imgeler kullanılmıştır. Steganografi yöntemi olarak LSB ve bu çalışmada verilen DLSB kullanılmıştır.

LSB Yöntemiyle imge steganografi daha önce çok sayıda çalışmada yer almıştır, bu nedenle bu yöntemin detaylı bir izahına bu çalışmada yeniden yer verilmemiştir. Bunun yerine, bu çalışmanın konusu olan DLSB'nin tanımlanabilmesi için yeterli olacak bir LSB tanımlaması yapılmıştır. Ancak LSB yönteminin detaylı bir incelemesi için Ref. [29] incelenebilir.

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

LSB yönteminde, örtü verisine ait bölümlerde her baytın en az anlamlı biti yerine gömü verisinin bitleri sırasıyla başlangıcından itibaren birer birer yerleştirilir. Burada her sekiz bitin en fazla bir biti değişikliğe uğratıldığından ve eğer değişiklik olmuşsa da baytın değişiklik yapılan biti en az anlamlı olanı olmasından dolayı, ortaya çıkan stego nesnesindeki değişimler insan tarafından algılanamaz boyuttadır. LSB, BMP (*Bitmap*) formatındaki imgeler üzerinde en çok bilinen ve en kolay uygulanan steganografi yöntemidir. Bir imgeye mesajın ('*gömü verisi' çalışma devamında 'mesaj' olarak verilmiştir*) yerleşiminde kullanılabilir imge baytının son 8. bitine mesajın yalnızca bir biti yerleştirilir, yani her bir mesaj biti için imgede bir bayt kullanılır. Örnek olarak Şekil 3A ile verilen $(66,114,75), (98,65,121), (83,97,78)$ piksel değerlerine, ASCII 'u' karakterinin karşılığı olan 117 değerinin ikilik tabandaki karşılığı olan 01110101 değerinin gömülmesi Şekil 3B ile örneklendirilmiştir. Bu temsilde toplam 3 pikselde yer alan toplam 9 ayrı bayt değerinin 8 adeti kullanılmıştır. Mesajın her bir bit değeri için, örtü imgenin bir bayt değerinin son bitinin sırasıyla mesaj bitleriyle değiştirildiği görülmektedir

BMP formatındaki bir imgenin $0,1,\dots,53$ konumlarındaki ilk 54 baytı başlık bilgilerine ayrılmıştır ve bu baytlar üzerine mesaj yerleştirilmez. Bu sebeple n_{bytes}^{Image} baytlık bir imge dosyasının toplam kullanılabilir bayt sayısı $n_{bytes}^{Image} - 54$ olur. $n_{bytes}^{Message}$ gizlenecek olan

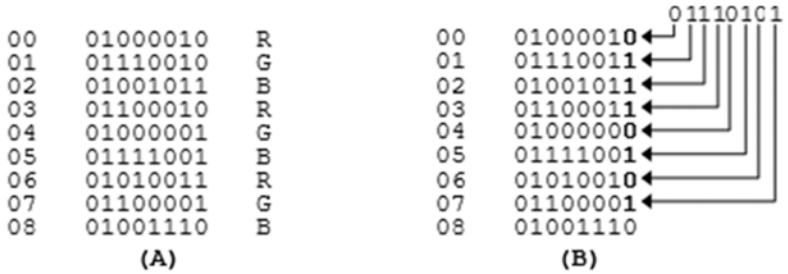
mesajdaki baytların sayısı olsun, o zaman mesajın toplam bit sayısı $n_{bits}^{Message} = 8\left(n_{bytes}^{Message}\right)$ olur. Ayrıca mesajın, mesaj çıkartma algoritmaları tarafından örtü imgesi içerisinden okunabilmesi için mesaj boyunun da örtü imgesi içerisine yerleştirilmesi gerekir. Burada $n^{Capacity}$-baytlık bir alan daha ayrılsın. $n^{Capacity}$-baytlık bu alanda gömülecek mesajın $0 \leq n_{bytes}^{Message} < 2^{n^{Capacity}}$ aralığında herhangi bir boyda olduğunu belirtmek mümkündür. Örneğin $n^{Capacity} = 16$ olsun ve $n_{bytes}^{Message} = 12345$ bayt boyunda bir mesaj gömülmek istensin, burada 12345'in ikilik tabandaki temsili (11000000111001) 14 basamaklı olduğundan $n^{Capacity} = 16$ yeterli olacaktır. Örtü imgesi içerisinde $n^{Capacity}$ seçimi keyfidir, diğer bir anlamda uygulayıcılar daha uzun veya kısa mesaj boyları için alan ayırabilirler. Ancak burada dikkat edilmesi gereken ayrılacak bu alanın örtü imgesinde meydana getireceği değişim oranıyla ilişkisidir. Ayrılacak alan ile birlikte örtü imgesindeki değişim oranı artar ve bu istatistiksel saldırılara karşı zaaf oluşturması nedeniyle istenen bir sonuç değildir.

Sonuç olarak 54-bayt başlık için ve $n^{Capacity}$-bayt mesaj boyu için olduğunda toplam kullanılabilir imge alanı $n_{free}^{Image} = n_{bytes}^{Image} - \left(54 + n^{Capacity}\right)$ olur. Buna göre $n_{bits}^{Message}$ bitlik bir mesajın n_{bytes}^{Image} baytlık bir imge içerisine yerleştirilmesi için $n_{free}^{Image} \geq 8\left(n_{bytes}^{Message}\right)$ olmalıdır. LSB ile bir imgeye mesajın yerleştirilmesi $54 + n^{Capacity}$ konumdaki bayttan başlar ve sırasıyla $54 + n^{Capacity}, 55 + n^{Capacity} + 1, \ldots, 54 + n^{Capacity} + n_{bits}^{Message}$ konumlarında devam eder. n_{free}^{Image} baytlık yazılabilir bir imge alanı $n_{free}^{Image}/8$ baytlık bir mesajı taşıyabilir. Bu durumda $n_{free}^{Image}/8 > n_{bytes}^{Message}$ olduğunda, $\left(n_{free}^{Image}/8\right) - n_{bytes}^{Message}$'lık bir alan boşluğu olur ve bu mesajın yazıldığı ilk $8\left(n_{bytes}^{Message}\right)$ imge baytında mesaj baytlarının yığılmasına neden olur. Bu da analiz bölümünde açıklanan Piksel Saldırısında büyük bir zafiyet oluşturur.

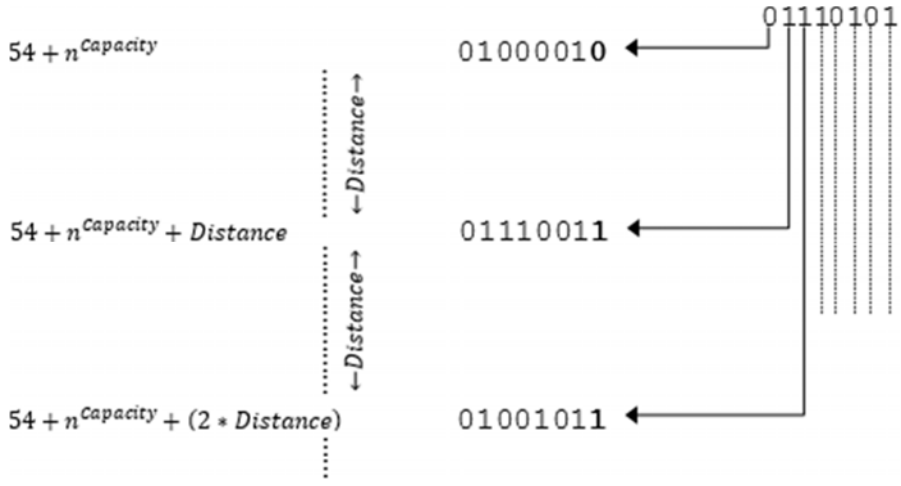
2.2. DLSB - Uzaklaştırılmış En Önemsiz Bit (Distanced Least Significant Bit)

DLSB, LSB ile birçok yönden aynı yöntemi kullanır. Burada yalnızca iki farklı yol izlenir. İlki mesaj bitlerinin imge baytlarına boşluksuz olarak yerleştirilmemesi. İkincisi imgenin $54,55, \ldots, 54 + n^{Capacity}$ konumundaki baytlarına mesajın boyunun değil, mesaj bitlerinin imgeye kaç bayt uzaklıklarla yerleştirileceğinin yazılması.

DLSB ile mesajın imge içerisine yerleştirilmesinde mesajın ve imgenin boyuna bağlı olarak, mesajın imge içerisine oranlı bir dağılımla / aralıkla yerleştirilmesi yolunu kullanılır. $n_{bits}^{Message} = 8\left(n_{bytes}^{Message}\right)$ bitlik bir mesajın n_{free}^{Image} baytlık bir imgeye gömülmesinde, her mesaj biti yerleşiminde imge baytları arasında $Distance = \left(n_{free}^{Image} - \left(n_{free}^{Image} \bmod n_{bits}^{Message}\right)\right)/n_{bits}^{Message}$ boşluk bırakılır. Bu $Distance$ değeri mesaj bitlerinin imge piksellerinde belirli bir bölümde yığılması yerine orantılı aralıklarla yerleştirilmesi anlamına gelir. Burada imgenin $54,55, \ldots, 54 + n^{Capacity}$ konumlarındaki her baytın son biti, $Distance$ değerinin ikilik tabandaki karşılığının sıradaki değiştirilir. Bu da LSB yönteminden farklı olarak, mesaj boyu yerine her mesaj bitinin önceki / sonraki mesaj bitiyle arasındaki uzaklığın imgeye yerleştirilmesidir. Burada imgeye $Distance$ değerinin aralıksız yerleştirildiğine, mesajın $Distance$ uzaklıkta yerleştirildiğine dikkat edilmelidir. Örnek olarak Şekil 4 ile $\left[54 + n^{Capacity}, 54 + n^{Capacity} + (2 * Distance)\right]$ aralığındaki örtü imgesi baytlarına, ASCII 'u' karakterinin karşılığı olan 117 değerinin soldan ilk üç biti olan 101 değerinin $Distance$-bayt aralıklar ile gömülmesi gösterilmiştir.



Şekil 3 LSB ile bir baytın gömülmesi (One byte embedding with LSB)



Şekil 4. DLSB ile bir baytın gömülmesi (One byte embedding with DLSB)

DLSB'nin bir algoritma olarak temsili için bit-tabanlı mantıksal operatörler kullanılmıştır. ≪, ≫, ve ∧ işaretleri, sırasıyla bit tabanında sola kaydırma, sağa kaydırma ve AND operatörlerini temsil ederler. Bu operatörlerin aritmetik karşılıkları $(a \gg b) \Leftrightarrow \left(\left(a - \left(a \bmod (2^b - 1)\right)\right)/2^b\right)$, $(a \ll b) \Leftrightarrow (2^b a)$, $(a \wedge 1) \Leftrightarrow (a \bmod 2)$ ve $(a \wedge 254) \Leftrightarrow \left(a - (a \bmod 2)\right)$ olarak verilebilir. O zaman DLSB için Şekil 5 ile bir imgeye mesajın gömülmesi ve çıkartılması için ilkel-algoritmalar verilmiştir.

DLSB yöntemi için ilkel-algoritmaların verildiği Şekil 5'deki Mesaj Gömme bölümü incelendiğinde yöntem daha kolay ve açık olarak anlaşılmaktadır. Burada 00-22 satır aralığında gömme adımları gösterilmiştir. Algoritma giriş olarak n_{bytes}^{Image} baytlık bir $Image$ imge dosyasını, $n_{bytes}^{Message}$ baytlık bir $Message$ mesajını, mesaj boyu kapasitesi için ayrılacak alan $n^{Capacity}$ değerini alır ve çıkış olarak n_{bytes}^{Image} baytlık $Output$ stego imgesini verir. 00. satırda örtü imgesinin kullanılabilir bayt sayısı n_{free}^{Image} ve 01. satırda mesajda yer alan bit sayısı belirlenmiştir. 05-06 satırlarda DLSB'nin temeli olan $Distance = \left(n_{free}^{Image} - \left(n_{free}^{Image} \bmod n_{bits}^{Message}\right)\right)/n_{bits}^{Message}$ değeri belirlenmiştir. Bu değer mesaja ait bitlerin, imge baytları içerisine kaçar bayt uzaklıkta yerleştirilerek dağıtılacağını belirlemek için kullanılır. 08-12 satır aralığında 8-bitlik bayt değerlerinden oluşan $Message$ dizisinin bitlerinden $MessageBits$ bit dizisi oluşturulmuştur. 13-15 satır aralığında $Distance$ değerinin ikilik tabandaki basamakları sırasıyla çıktı olacak stego imgenin $\left[54, \left(54 + n^{Capacity}\right) - 1\right]$ aralığı konumlarındaki baytların en önemsiz bitlerine yerleştiriliştir. 16-21 satır aralığında mesajın bit değerleri örtü imgenin $\left[\left(54 + n^{Capacity}\right), \left(54 + n^{Capacity} + n_{bytes}^{Message}\right) - 1\right]$ aralığında ve birbirlerinden $Distance$ uzaklıkta yer alan bayt değerlerinin en önemsiz bitlerine yerleştirilmiştir. Son olarak 22. satırda DLSB stego imgesi $Output$ çıkışı elde edilmiştir.

Elde edilen stego imgeden, mesajın çıkartılması da Şekil 5 Mesaj Çıkartma bölümünde verilmiştir. Burada 00. satırda mesajın hangi bayttan itibaren yerleştirildiği belirlenmiştir. 02-04 satır aralığında mesaja ait bitlerin, stego imgesine kaçar bayt uzaklıkta yerleştirildiğini belirten $Distance$ değeri bulunmuştur. 05-06 satır aralığında önce mesajın kaç bit sonra kaç bayt olduğu bulunmuştur. 08-14 satır aralığında stego imgesine $Distance$ bayt aralıklarla

DLSB – Mesaj Gömme

Giriş: n_{bytes}^{Image} baytlık $Image$ dizisi; $Message$ boyu kapasitesi için ayrılacak alan $n^{Capacity}$; $n_{bytes}^{Message}$ baytlık $Message$ dizisi.
Çıkış: $n_{bytes}^{Message}$ baytlık $Message$ gömülü n_{bytes}^{Image} baytlık $Output$ imge dizisi.

```
00: n_free^Image := n_bytes^Image − (54 + n^Capacity);
01: n_bits^Message := 8 * n_bytes^Message;
02: if n_free^Image < n_bits^Message then
03:     return "Insufficient image size ...";
04: end if
05: Distance := n_free^Image − (n_free^Image mod n_bits^Message);
06: Distance := Distance/n_bits^Message;
07: Output := Image;
08: for i := 0 to n_bytes^Message − 1 do
09:     for j := 0 to 7 do
10:         MessageBits[(8 * i) + j] := ⤸
                (Message[i] ≫ (7 − j)) ∧ 1;
11:     end for
12: end for
13: for i := 0 to (n^Capacity − 1) do
14:   Output[54 + i] := (Output[54 + i] ∧ 254) + ⤸
        ((Distance ≫ ((n^Capacity − 1) − i)) ∧ 1);
15: end for
16: for i := 0 to n_bytes^Message − 1 do
17:     for j := 0 to 7 do
18:         k := (54 + n^Capacity) ⤸
              +(((8 * i) + j) * Distance);
19:         Output[k] := (Output[k] ∧ 254) + ⤸
                        MessageBits[(8 * i) + j];
20:     end for
21: end for
22: return Output;
```

DLSB – Mesaj Çıkartma

Giriş: n_{bytes}^{Image} baytlık $Image$ dizisi; $Message$ boyu kapasitesi için ayrılmış alan $n^{Capacity}$.
Çıkış: $n_{bytes}^{Message}$ baytlık $Message$ dizisi.

```
00: n_free^Image := n_bytes^Image − (54 + n^Capacity);
01: Distance := 0;
02: for i := 0 to (n^Capacity − 1) do
03:     Distance := Distance + ⤸
    ((Image[54 + i] ∧ 1) ≪ ((n^Capacity − 1) − i));
04: end for
05: n_bits^Message := n_free^Image − (n_free^Image mod Distance);
06: n_bits^Message := n_bits^Message /Distance
07: n_bytes^Message := n_bits^Message /8;
08: k := 54 + n^Capacity;
09: for i := 0 to n_bytes^Message − 1 do
10:     for j := 0 to 7 do
11:         Message[i] := Message[i] + ⤸
                ((Image[k] ∧ 1) ≪ (7 − j));
12:         k := k + Distance;
13:     end for
14: end for
15: return Message;
```

Şekil 5. DLSB – Mesaj gömme ve çıkartma için ilkel-algoritmalar
(DLSB – Pseudo-algorithms for message embedding and extracting)

yerleştirilen mesaj bitlerinden $Message$ bayt dizisi oluşturulmuştur. Son olarak 15. satırda gömü verisi olan $Message$ bayt dizisi mesajı elde edilmiştir.

3. Analizler (Analyses)

Steganografi yöntemlerinin güvenlik analizlerinde en çok kullanılan steganaliz araçları Düzenli-Tekil (*RS – Regular-Singular*), Çiftliler Örneği (*Sample of Pairs*), Ki-Kare (*Chi-Square*), Birincil Kümeler (*Primary Sets*), En Büyük Sinyal Gürültü Oranı (*PSNR - Peak Signal-to-Noise Ratio*) ve Piksel Saldırısıdır. Literatürde yer alan steganografi yöntemleri çeşitli sıkıştırma yöntemleriyle imgeye yerleştirilecek mesajı olabildiğince kısaltarak, imgede en az bilgi değişimi gerçekleştirmeyi ve istatistiksel steganaliz yöntemlerine direnç sağlamayı amaçlar. Ancak mesaj ne kadar sıkıştırılırsa sıkıştırılsın, imgeye yerleşiminin belirli bir bölgede yığılması nedeniyle Piksel Saldırısına karşı zaaf oluşturur. Bu çalışma ile verilen DLSB mesaj yerleşimi için bir temel yöntemdir. Tek başına istatistiksel steganaliz yöntemlerine karşı bir direnç sağlamaz. Asıl direnç özelliği Piksel Saldırısına karşıdır. Bu yönüyle literatürde yer alan diğer steganografi yöntemlerinde de kullanılabilecek bir mesaj yerleşim düzeni sağlar. Ancak okurlara ve sonraki çalışmalara kaynak olması amacıyla sıkıştırmasız ve sıkıştırmalı olarak DLSB yönteminden elde edilecek sonuçların detaylı steganalizi bölüm 3.2 ile verilmiştir. DLSB yönteminin gömme kapasitesi veya hız performansı için başarı amacı yoktur, bu DLSB yöntemini kullanarak geliştirilebilecek sıkıştırma fonksiyonlu yöntemlerin incelemesi gereken bir konudur. Bu bölümde gerçekleştirilen analiler için MATLAB R2012a [30] ile birlikte gelen 3060×2036 boyutundaki concordaerial.png imge 512×340 boyutunda A.bmp, 4096×2048 boyutundaki imagefusiondemo_01.png imge 512×384 boyutunda B.bmp ve 512×256 boyutundaki wpeppers2.png imge 512×256 boyutunda C.bmp olarak 24-bit BMP formatına (*.bmp*) dönüştürülerek kullanılmıştır. Ayrıca hem LSB hem de DLSB için $n^{Capacity} = 16$ olarak seçilmiştir.

3.1. Piksel Saldırısı (Pixel Attack)

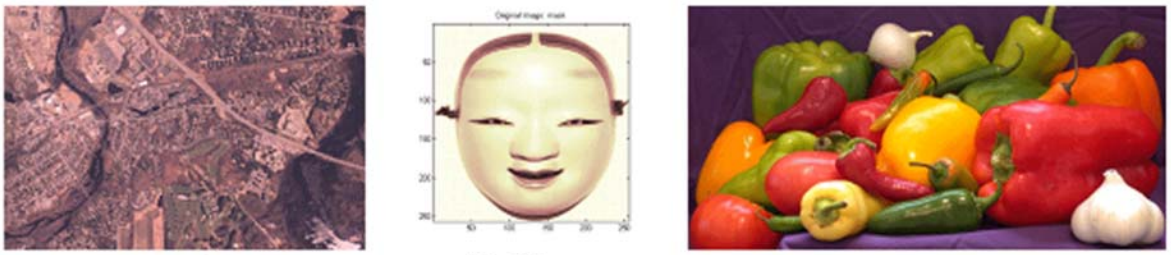
Piksel Saldırısı her pikselin LSB değerine bağlı olarak, değerinin arttırılması veya azaltılması ile gerçekleştirilir. Burada saldırılacak imge dosyasının bütün pikselleri baştan sona taranır. Pikselin kanal renginin değeri son biti 0 ise 0, 1 ise 255 olarak değiştirilerek LSB bitlerinin belirginleştirilmesi sağlanır. Bunu bir (Eş. 1) fonksiyonu olarak tanımlayabiliriz.

$$PixelAttack(a) = \begin{cases} 0 & ,0 = a \bmod 2 \\ 255 & ,1 = a \bmod 2 \end{cases} \quad (1)$$

BMP formatında piksel değerlerinin 54. konumdaki bayttan başladığını hatırlayalım. O zaman bir imge üzerinde bir saldırı $Image[i] = PixelAttack(Image[i])$,$54 \le i < n_{bytes}^{Image}$ ile gerçekleşebilir. Analizler için kullanılan farklı boyutlardaki A.bmp, B.bmp ve C.bmp orijinal imgeleri Şekil 6A, Şekil 6B ve Şekil 6C ile verilmiştir. Bu imgelere Piksel Saldırısı uygulandığında elde edilen imgeler Şekil 7A, Şekil 7B ve Şekil 7C verilmiştir.

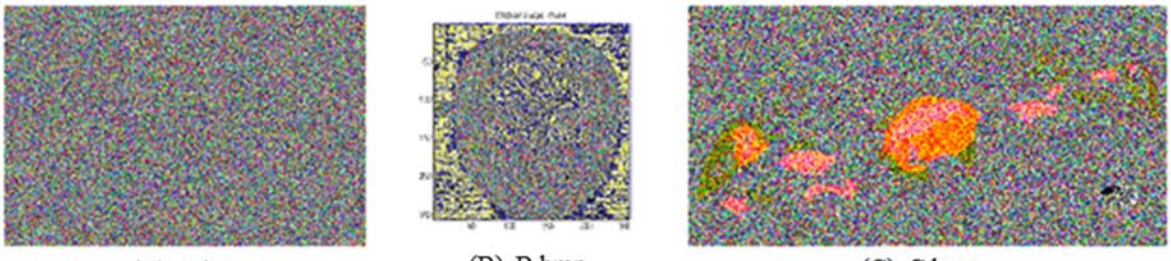
Bölüm 2.1 ile açıklandığı gibi ilk $n^{Capacity} = 16$ olduğunda ilk $54 + n^{Capacity}$ bayt değeri imgenin başlık ve mesajın toplam bayt sayısı için rezerve edildiğinden kullanılabilir toplam bayt sayısı $n_{bytes}^{Image} - \left(54 + n^{Capacity}\right)$ olur. Her 8-bitlik baytın yalnızca son bit değeri kullanılacağından, imgeler içerisine gömülebilecek mesaj boyutu $n_{bytes}^{Message} \le \left(n_{bytes}^{Image} - \left(54 + n^{Capacity}\right)\right)/8$ koşuluna sahip olmalıdır. Tekrarlanabilir ve güvenilir bir test için gömülecek mesaj olarak CrypTool 1.4.41 [31] yazılımı ile gelen ve "*CrypTool\words\cracklib-words*" konumunda verilen doğal dil kelimeleri kullanılmıştır. Burada i konumundaki 8-bitlik harf $Words_i$ olarak gösterilmiştir.

Öncelikle $\left(n_{bytes}^{Image} - \left(54 + n^{Capacity}\right)\right)/8$ baytlık alanın $\approx$ %5 i olan harf dizisi $TestMessageA_i = Words_i$,$0 \le i < (5/100)\left(\left(n_{bytes}^{Image} - \left(54 + n^{Capacity}\right)\right)/8\right)$ olarak seçilmiştir. Şekil 8A, Şekil 8B ve Şekil 8C ile LSB yöntemiyle mesaj gömme sonrası elde edilen imgelere Piksel Saldırısı gerçekleştirilmesi sonucu imgelerin alt kısımlarında yerleştirilen mesajın izleri açık olarak görülmektedir. Daha büyük bir mesaj boyu olan $\left(n_{bytes}^{Image} - \left(54 + n^{Capacity}\right)\right)/8$ baytlık alanın $\approx$ %50 si olan harf dizisi



(A) A.bmp (B) B.bmp (C) C.bmp

Şekil 6 Orjinal İmgeler (Original Images)



(A) A.bmp (B) B.bmp (C) C.bmp

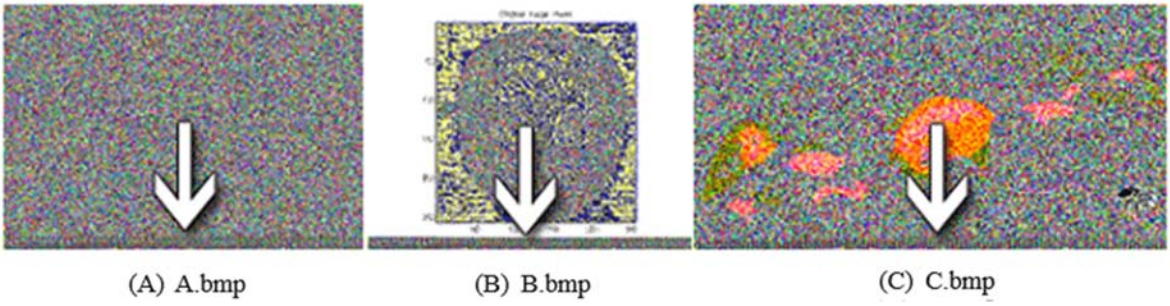
Şekil 7 Orijinal İmgeler, Piksel Saldırısı (Orijinal Images - Pixel Attack)

$TestMessageB_i = Words_i, 0 \leq i < (50/100)\left(\left(n_{bytes}^{Image} - \left(54 + n^{Capacity}\right)\right)/8\right)$ kullanılarak Piksel Saldırısı uygulandığında Şekil 9A, Şekil 9B ve Şekil 9C ile imgelerin alt kısımlarında mesaj izleri çok daha açık olarak görülmektedir.

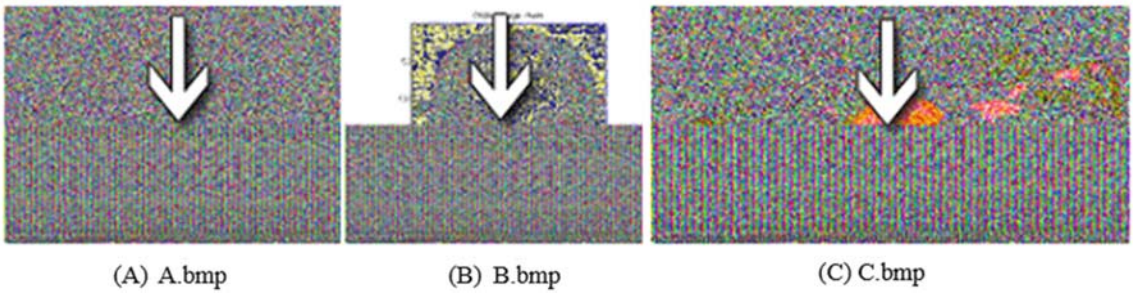
$(5/100)\left(\left(n_{bytes}^{Image} - \left(54 + n^{Capacity}\right)\right)/8\right)$ baytlık harf dizisi ve DLSB yöntemiyle elde edilen stego imgelere Piksel Saldırısı gerçekleştirildiğinde elde edilen imgeler Şekil 10A, Şekil 10B ve Şekil 10C incelendiğinde veri izlerinin gizlendiği görülmektedir. Benzer olarak $(50/100)\left(\left(n_{bytes}^{Image} - \left(54 + n^{Capacity}\right)\right)/8\right)$ baytlık harf dizisinin kullanıldığı Şekil 11A, Şekil 11B ve Şekil 11C için de veri izlerinin başarılı bir şekilde gizlendiği görülmektedir. Burada gerçekleştirilen testler DLSB yönteminin Piksel Saldırısına karşı başarılı bir direnç oluşturduğunu göstermiştir.

3.2. İstatistiksel Analiz (Statistical Analysis)

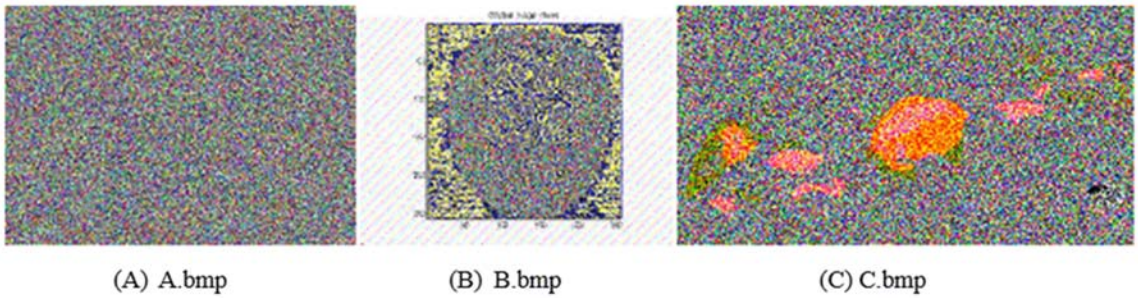
Devamda verilen ek steganaliz yöntemleri DLSB yönteminin incelenmesi ve test edilmesi için kullanılmıştır. Bu analiz



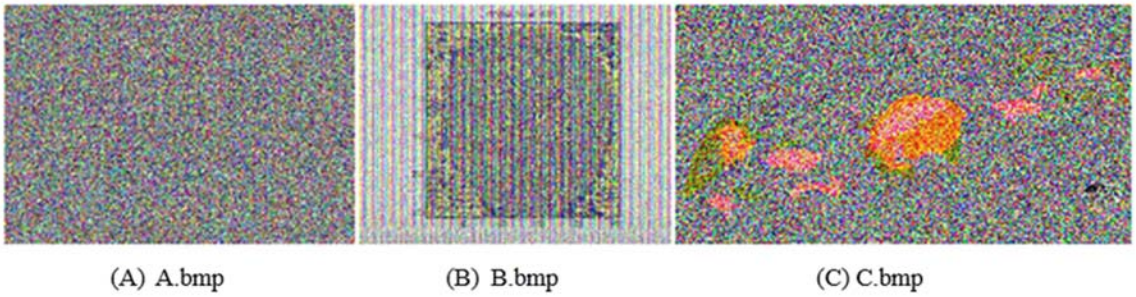
Şekil 8 LSB - %5 bayt mesaj - Piksel Saldırısı (LSB - %5 byte message – Pixel Attack)



Şekil 9 LSB - %50 bayt mesaj - Piksel Saldırısı (LSB - %50 byte message - Pixel Attack)



Şekil 10 DLSB - %5 bayt mesaj - Piksel Saldırısı (DLSB - %5 byte message - Pixel Attack)



Şekil 11 DLSB - %50 bayt mesaj - Piksel Saldırısı (DLSB - %50 byte message - Pixel Attack)

yöntemlerinin uygulanması ile ilgili detaylı teknik içeriğe verilen referanslardan ulaşılabilir. En Büyük Sinyal Gürültü Oranı (*PSNR - Peak Signal Noise Ratio*) steganografi performans ölçümünde sıkça kullanılır [32,33]. En Büyük Sinyal Gürültü Oranı ile ilgili deneysel sonuçlar örtü ve stego imgesi arasındaki 66'dan daha büyük değerlerin İnsan Görülebilir Sistemi (*HVS - Human Visible System*) için şüphe oluşturduğunu göstermiştir [34]. RS analizi [35], LSB için düzenli ve tekil grupların sayısındaki farklılıkların teşhisini kullanarak gri-ölçekli ve renkli imgelerdeki rastlantısal dağılımları denetler. Örnek Çiftliler (*Sample Pairs*) analizi [36], çoklu-düzen izleri olarak adlandırılan örnek çiftlilerin seçilen çoklu-düzen durumları olan sonlu durum makinesini temel alır. Ki-Kare (*Chi-Square*) analizi [28], LSB gömme işlemi boyunca değiş-tokuş edilen Değerler Çiftlilerinin (*PoV's – Pair of Values*) istatistiksel analizidir. Değerler Çiftlileri bir LSB nesnesindeki ikilik değerlerin gruplarıdır. Birincil Kümeler (*Primary Sets*) [37], bir imgedeki bir dizi piksellerin kesinliğiyle ilgili istatistiksel özdeşliği temel alır. Farksal Histogram (*Difference Histogram*) analizi [38], bir imgenin histogramı üzerinde istatistiksel saldırıdır, en önemsiz diğer bütün bit yüzeyleri arasındaki korelasyonu ölçer. Burada gerçekleştirilen analizlerde En Büyük Sinyal Gürültü Oranı için $x \geq 66$; RS, Ki-Kare, Değerler Çiftlileri ve Birincil Kümeler için $x \leq 0{,}2$ istenen en sonuçlardır.

Tablo 1 ile orijinal A.bmp, B.bmp ve C.bmp imgelerine ait PSNR, Primary Sets, Chi-Square, Sample of Pair ve RS Analysis sonuçları verilmiştir.

Tablo 1 İstatistiksel Sonuçlar – Orijinal İmgeler
(Statistical Results – Original Images)

	A.bmp	B.bmp	C.bmp
		PSNR	
Orj.	∞	∞	∞
		Primary Sets	
Orj.	0,0163	NaN	0,0190
		Chi-Square	
Orj.	0,0	0,1507	0,0345
		Sample of Pair	
Orj.	0,0083	0,0355	0,0200
		RS	
Org.	0,0086	0,0673	0,0370

LSB ve DLSB imgeleri için elde ede edilen PSNR, Primary Sets, Chi-Square, Sample of Pair ve RS Analysis sonuçları Tablo 2 ile verilmiştir. PSNR değerleri incelendiğinde LSB ve DLSB arasında anlamlı bir fark olmadığı görülmektedir. Primary Sets, Chi-Square, Sample of Pair ve RS Analysis değerleri incelendiğinde LSB yönteminin DLSB yönteminden daha küçük değerler aldığı görülmektedir. Ancak burada DLSB sadece bir veri yerleşim yöntemi ve Piksel Saldırı direnci amaçladığından bu sonuçların tek başına bir anlamı olmayacaktır.

Yukarıda verilen sonuçlara göre DLSB'nin LSB'den daha güvenli ve/ya güvensiz olduğunu söylemek mümkün değildir. Burada kullanılan A.bmp, B.bmp ve C.bmp imgelerinin piksellerinde yer alan RGB değerleri ve bit yayılımları bu sonuçları etkilemektedir. Nitekim Tablo 2'de yer alan Sample of Pair testinde A.bmp imgesinin %5 ve %50 bayt mesaj yerleşimi için, ve RS testinde A.bmp imgenin %50 bayt mesaj yerleşimi için DLSB'nin daha güvenli olduğu gözlenmiştir. Bu da bizi "*…bir deney yeterince tekrarlandığında teorik olasılık oranı ile uygulamadaki olasılık oranı arasındaki fark çok az olur; hatta deney sayısı sonsuz kez tekrarlanırsa, bu fark sıfır olur…*" [39] olarak bilinen büyük sayılar kuralına göre örneklem sayısını arttırmaya yönlendirir. Bu sebeple SIPI İmge Veri tabanından [40] rastlantısal olarak seçilmiş 10^5 adet imge dosyası orijinal dikey ve yatay boyutları korunarak BMP formatına dönüştürülmüş ve örtü imgesi olarak kullanılmıştır. Her örtü imgesi kapasitesinin hem %5'i hem de %50'si oranında rastlantısal değerlerden oluşan mesaj kullanılmıştır. Bu testten elde edilen sonuçlar Tablo 3 ile verilmiştir. Tablo 2 ve Tablo 3 değerleri karşılaştırıldığında çoklu örneklemlerden elde edilen test sonuçlarının ortalaması için DLSB'nin LSB'den daha başarılı olduğu ortaya çıkmaktadır. Burada PSNR sonuçlarının büyük olması örtü imgesindeki görsel bozulmanın az olduğunu göstermektedir. Ayrıca Sample of Pair testinde %50 baytlık mesaj için elde edilen sonuç haricinde diğer bütün testler için $< 2{,}0$ altında değerler aldığı ve güvenlik kriterini karşıladığı görülmektedir. Diğer taraftan Sample of Pair testinde %50 baytlık mesajda $0{,}2325$ değerine ulaşarak güvenlik kritik değerinin aşıldığı ancak $2{,}2785$ değerine ulaşan LSB'den daha güvenli sonuç verdiği görülmektedir.

DLSB yönteminin bir sıkıştırma yöntemiyle birlikte kullanılması durumunda elde edilecek sonucu örneklendirmek için %5 ve %50 baytlık mesajları Info-Zip yöntemi [41] ile sıkıştırdığımızda DLSB

Tablo 2 İstatistiksel Sonuçlar – LSB ve DLSB İmgeleri (Statistical Results – LSB and DLSB Images)

	%5 bayt			%50 bayt		
	A.bmp	B.bmp	C.bmp	A.bmp	B.bmp	C.bmp
			PSNR			
LSB	64,1462	63,4942	64,1301	54,1641	54,0087	54,1403
DLSB	64,1869	63,7977	64,1477	54,1518	54,0226	54,1549
			Primary Sets			
LSB	NaN	0,0597	0,0462	NaN	0,3176	NaN
DLSB	NaN	0,0722	0,0872	NaN	NaN	NaN
			Chi-Square			
LSB	0,1507	0,0	0,0345	0,1512	0,0	0,0345
DLSB	0,1717	0,0	0,0346	0,7123	0,0304	0,2318
			Sample of Pair			
LSB	0,0833	0,0448	0,0584	0,3491	0,2500	0,3569
DLSB	0,0772	0,0649	0,0739	0,2744	0,4161	0,3727
			RS			
LSB	0,0998	0,0422	0,0687	0,3511	0,2548	0,3362
DLSB	0,1041	0,0648	0,0856	0,3489	0,4235	0,3844

Tablo 3 İstatistiksel Sonuçlar – LSB ve DLSB (Statistical Results – LSB and DLSB)

	%5 bayt	%50 bayt
	PSNR	
LSB	64,1746	54,0747
DLSB	64,2033	56,1872
	Primary Sets	
LSB	0,0673	0,2377
DLSB	0,0407	0,1720
	Chi-Square	
LSB	0,1392	0,0671
DLSB	0,0743	0,0525
	Sample of Pair	
LSB	0,0654	0,2785
DLSB	0,0311	0,2325
	RS Analysis	
LSB	0,0635	0,2602
DLSB	0,0458	0,1914

Tablo 4 İstatistiksel Sonuçlar – DLSB ile Info-Zip Sıkıştırması (Statistical Results – DLSB with Info-Zip Compression)

	%5 bayt			%50 bayt		
	A.bmp	B.bmp	C.bmp	A.bmp	B.bmp	C.bmp
			PSNR			
DLSB	68,2641	68,2281	65,8547	61,5219	61,2279	58,6415
			Primary Sets			
DLSB	NaN	0,0164	0,0505	NaN	0,0163	0,2503
			Chi-Square			
DLSB	0,1590	0,0	0,0352	0,1636	0,0	0,0345
			Sample of Pair			
DLSB	0,0558	0,0215	0,0536	0,1310	0,1132	0,1831
			RS Analysis			
DLSB	0,0811	0,0214	0,0692	0,1534	0,1148	0,2123

için elde edilen sonuçlar Tablo 4 ile verilmiştir. Burada DLSB yönteminin orijinal imge değerlerine çok yakın değerler verdiği görülmektedir ve bu DLSB yerleşim yöntemini kullanacak geliştirmelerde başarılı sonuçlar elde edilebileceğini göstermektedir.

4. Sonuçlar (Conclusions)

LSB yöntemiyle Bitmap imge dosyaları üzerinde gerçekleşen steganografi Piksel Saldırısına karşı direnç gösterememektedir. Çeşitli sıkıştırma yöntemleriyle mesajın sıkıştırılmasına dayanan diğer steganografi yöntemleri de bu saldırıdan etkilenmektedir. Bu çalışmada verilen DLSB yönteminde, mesaj bitlerinin basit bir şekilde imge içerisine orantılı olarak dağıtılması sağlanmıştır. Gerçekleştirilen testlerde LSB yönteminin piksel saldırısına karşı zafiyeti olduğu ve DLSB yönteminin mesajın imge içerisinde belirli bir alanda yığılmaması sayesinde piksel saldırısına karşı dirençli olduğu görülmüştür. PSNR, Primary Sets, Chi-Square, Sample of Pair ve RS analizi testlerinde de DLSB'nin başarılı sonuçlar elde ettiği görülmüştür. Ayrıca DSLB'nin bir sıkıştırma yöntemiyle beraber kullanılabileceği ve bu durumunda analiz sonuçlarının daha başarılı olabileceği görülmüştür. DLSB, LSB yöntemine karşı gerçekleştirilen Piksel Saldırısına karşı koruma sağlayan ilk yöntemdir.

Kaynaklar (References)

1. Şahin A., Görüntü Steganografide Kullanılan Yeni Metodlar ve Bu Metodların Güvenilirlikleri, Doktora Tezi, Trakya Üniversitesi, Trakya, Türkiye, 2007.
2. Kahn D., The Codebreakers, Macmillan Publishing Company, New York, 1967.
3. Fridrich J., Steganography in Digital Media: Principles, Algorithms and Applications, Cambridge Unv. Press, Cambridge, 2009.
4. Sharp T., An Implementation of Key-Based Digital Signal Steganography, Lecture Notes in Computer Science, Springer, Berlin, 2001.
5. Mielikainen J., LSB matching revisited, IEEE Signal Processing Letters, 13 (5), 285-287, 2006.
6. Chan, On using LSB matching function for data hiding in pixels, Fundamenta Informaticae, 96 (1), 49-59, 2009.
7. Tian J., Reversible data embedding using a difference expansion, IEEE Transactions on Circuits and Systems, 13 (8), 890-896, 2003.
8. Alattar A.M., Reversible watermark using the difference expansion of a generalized integer transform, IEEE Transactions on Image Processing, 13 (8), 1147-1156, 2004.
9. Chang C.C., Chou Y., Kieu T.D., Information hiding in dual images with reversibility, Third International Conference on Multimedia and Ubiquitous Engineering, Qingdao, Çin, Haziran 2009.
10. Lu T., Tseng C., Wu J., Dual imaging-based reversible hiding technique using LSB matching, Signal Processing, 108, 77-89, 2015.
11. Ker A.D., Quantitive evaluation of pairs and RS steganalysis, Security, Steganography, and Watermarking of Multimedia Contents VI, Haziran 2004.
12. Wu N.I., Tsai W., A steganographic method for images by pixel-value differencing, Pattern Recognition Letters, 9-10 (24), 1613-1626, 2003.
13. Wang C.M., Wu N.I., Tsai C.I., Hwang M.S., A high quality steganographic method with pixel-value differencing and modulus function, Journal of Systems and Software, 81 (1), 150-158, 2008.
14. Fridrich J., Soukal D., Matrix Embedding for Large Payloads, EE Transactions on Information Forensics and Security, 3 (1), 390-395, 2006.
15. Kurtuldu O., Arıca N., A new steganography method using image layers, 23rd International Symposium on Computer and Information, İstanbul, Türkiye, 27-29 October 2008.
16. Wu H., Wang H., Hu Y., Zhou L., Efficient reversible data hiding based on prefix matching and directed LSB embedding, Digital-Forensics and Watermarking: 13th International Workshop, Taipei, Taiwan, 1-4 Ekim 2014.

17. Huang F., Zhong Y., Huang J., Improved algorithm of edge adaptive image steganography based on LSB matching revisited algorithm, Lecture Notes in Computer Science, 8389, 19-31, 2014.

18. Sabeti V., Samavi S., Shirani S., An adaptive LSB matching steganography based on octonary complexity measure, Multimedia Tools and Applications, 3, (64), 777-793, 2013.

19. Jain R., Kumar N., Efficient data hiding scheme using lossless data compression and image steganography, International Journal of Engineering Science and Technology, 8 (4), 3908-3915, 2012.

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

21. Ozer H., Avcıbas I., Sankur B., Memon N., Steganalysis using image quality metrics, IEEE Trans. Image Process., 2 (12), 221-229, 2003.

22. Ozer H., Avcıbas I., Sankur B., Memon N., Steganalysis of audio based on audio quality metrics, Security and Watermarking of Multimedia Contents V, 5020 of Proceedings of SPIE, 55-66, Ocak 2003.

23. Batagelj V., Bren M., Comparing resemblance measures, Proc. International Meeting on Distance Analysis (DISTANCIA'92), Haziran 1992.

24. Sokal R.R., Sneath P.H.A., Numerical Taxonomy, The Principles and Practice of Numerical Classification, W. H. Freeman, San Francisco, California, A.B.D., 1973.

25. Ojala T., Pietikàinen M., Harwood D., A comparative study of texture measures with classification based on feature distributions, Pattern Recognit., 29 (1), 51-59, 1996.

26. Christy A.S., Pairs of Values and the Chi-squared Attack, Department of Mathematics, Iowa State University, 2005.

27. Katzenbeisser S., Information Hiding: techniques for steganography and digital watermarking, Artech House, Boston, M.A., A.B.D., 2000.

28. Westfeld A. and Pfitzmann A., Attacks on steganographic systems, Information Hiding, 61-76, 2000.

29. Ertürk İ., Çetin Ö., Yalman Y., Akar F., Veri Gizleme, Beta Basım Yayım, İstanbul, Türkiye, 2014.

30. The MathWorks Inc. MathlLab. http://www.mathworks.com, Erişim tarihi Aralık 5, 2022.

31. CrypTool. CrypTool 1.4.41. http://www.cryptool.org, Erişim tarihi Aralık 5, 2022.

32. Manimurugan S., Saad Al-Mutairi, Novel secret image hiding technique for secure transmission, Journal of Theoretical & Applied Information Technology, 95.1 (166), 2017.

33. Sharifara M.S.M.R.A. and Morteza B., A novel approach to enhance robustness in digital image watermarking using multiple bit-planes of intermediate significant bits, International Conference on IEEE, 2013.

34. Mohammed M.H., Rahim M.S.M., Image steganography based on odd/even pixels distribution scheme and two parameters random function, Journal of Theoretical and Applied Information Technology, 95 (22), 5978-5986, 2017.

35. Fridrich J., M. Goljan,, Du R., Reliable detection of LSB steganography in color and grayscale images, Proceedings of the 2001 workshop on Multimedia and security: new challenges, 27-30, 2001.

36. Dumitrescu S., Xiaolin W., Wang Z., Detection of LSB steganography via sample pair analysis, International workshop on information hiding, Berlin, Heidelberg, 355-372, 2002.

37. Dumitrescu S., Wu X., Memon N., On steganalysis of random LSB embedding in continuous-tone images, Proceedings. International conference on image processing, 641-644, 2002.

38. Zhang T., Xijian P., Reliable detection of LSB steganography based on the difference image histogram, 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, III-545, 2003

39. Eren, Ş., Razbonyalı, M., Şengonca, H., Okatan, A., Yazıcı, A., Erten, M., Arsan, T., Duru, N., Çölkesen, T., Turan, A., Uğurkaya, C., Sarıdoğan, E., Acar, A., Bilgisayar Mühendisliğine Giriş, Papatya Yayıncılık Eğitim, İstanbul, Türkiye, 2020.

40. University of Southern California, Signal and Image Processing Institute. SIPI Image Database. http://sipi.usc.edu/database/. Erişim tarihi Aralık 5, 2022 Info-ZIP group. Info-Zip. http://www.info-zip.org. Erişim tarihi Aralık 5, 2022