

# Stenografi

---

Ağ ve Bilgi Güvenliği Dersi – Necmettin ÇARKACI



# Stenografi

- ⌚ Eski yunanca gizlenmiş bilgi anlamındadır.
- ⌚ Alfabenin harfleri, noktalama işaretleri, kelimeleri yerine semboller ve kısaltmalar kullanan **çabuk yazma; not tutma sistemi**.
- ⌚ İlk zamanlar katipler tarafından kullanılan bir yöntemdi.
- ⌚ Günümüzde hala aktif olarak kullanılıyor.
- ⌚ TBMM tutanakları. [TBMM Stenografi dokümanı](#).
- ⌚ Bilginin gizlenerek iletilmesi amacıyla da kullanılıyor.
- ⌚ Amaç, mesajı şifrelemek değil varlığını saklamak.





# Stenografi

---

- ⌚ Günümüzde cep telefonlarının çekilen 1600 \* 1200 çözünürlükteki bir resmin içine yaklaşık 2 mb'a kadar veri saklanabilir. (wikipedia)
- ⌚ Resim gizleme için fazla veri gizleme metodu bulunmaktadır. Bu yöntemler arasında en iyi en kötü ayrımı yapamayız! (wikipedia)
- ⌚ Günümüzde stenografi kolayca tespit edebilen yöntemler mevcut değil.
- ⌚ Tespit için genelde dosya büyüklüğü vb. değişikliklere bakılıyor. Dosya büyüklüğünü azaltan araçlar ve yöntemler günümüzde mevcut.
- ⌚ Daha çok istatiksel yöntemler kullanılıyor.
- ⌚ Resmin orijinal kopyası sahip olmak büyük avantaj sağlıyor.
- ⌚ Bazı resimlerde az bilgi çok fazla bozulmaya neden olabiliyor.



# Stenografi Çeşitleri

---

- ⌚ **Metin içine** metin gizleme (Null chipher)
- ⌚ **Program içine** doküman / program gizleme Alternate Data Streams(ADS)
- ⌚ **Görüntü içine** doküman / program gizleme
- ⌚ **Ses içine** doküman / program gizleme
- ⌚ **Video içine** doküman / program gizleme



# Metin Stenografi (Null Ciphering)

- ⌚ Mesaj başka bir mesajın içine belirli kurallar çerçevesinde saklanıyor.
- ⌚ Önceden belirlenmiş bir «codeword» var.
- ⌚ Genelde kısa mesajların iletiminde kullanılıyor.
- ⌚ Mesajın büyüklüğü iletilen taşıyan doküman büyüklüğünü de artırıyor.
- ⌚ Halka açık gazete, dergi, blog vb. yazıları üzerinden iletim sağlanıyor.

**Mesaj :** Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-product, ejecting suets and vegetable oils.

**Gizli Mesaj :** Pershing sails from NY June 1

1. Dünya Savaşında Almanlar tarafından kullanıldı



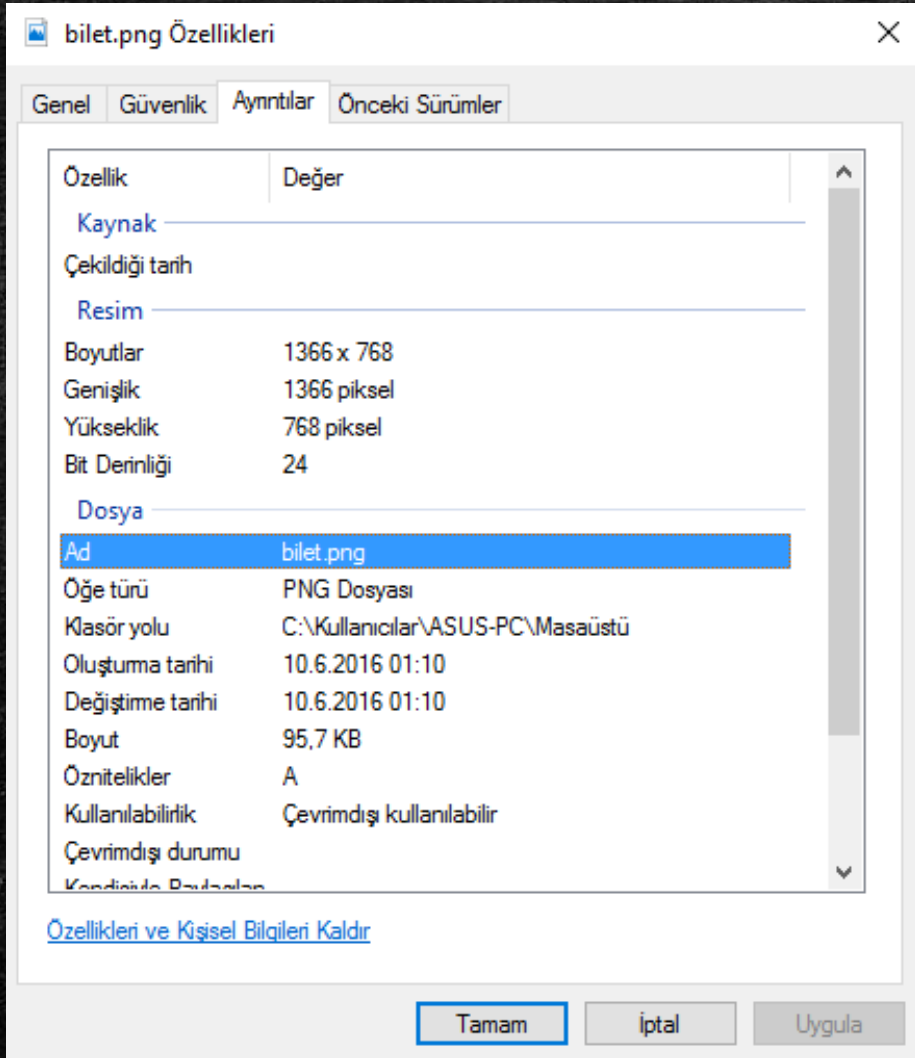
# ADS (Alternate Data Streams)

---

- ⌚ Microsoft dosya sistemi NTFS (New Technology File System), Mac HFS (Hierarchical File System) uyumluluk gösterir .
- ⌚ HFS dosyayı ikiye böler, dosya ayarlarını gizli olarak dosyanın içinde tutar.
- ⌚ Bir dosya içine başka dosya gizlemesine olanak sağlar.
- ⌚ 1 kb'lık bir metin belgesinin içerisine boyutunu kesinlikle değiştirmeden 100'lerce megabyte veri saklamak mümkün.
- ⌚ Veriyi sadece tam olarak hangi isimle nerde olduğunu bilen kişi çalıştırabilir.
- ⌚ Dosyalar herhangi bir fat32 sisteme taşınırsa gizli dosyalar kaybolur.



# ADS (Alternate Data Streams)



- ⌚ Dosya metadata bilgileri ADS ile tutulur.
- ⌚ Bir dosya için sınırsız sayıda ve boyutta bağlantı oluşturulabilir.
- ⌚ Dosya yanı sıra klasör ve dosya sisteminde bağlantı oluşturulabilir.
- ⌚ Sadece metin dosyaları değil, ses, görüntü ve çalıştırılabilir dosyalar içinde bağlantı oluşturulabilir.
- ⌚ Virüsler bu şekilde dosyalarla bağlanabilir.
- ⌚ Windows tarafından ADS görüntülemek için herhangi bir komut yada program yok.



# Uygulama - ADS

---

- ① **%windir%\system32\notepad.exe belge.txt** komutuyla bir doküman oluşturun ve içine herhangi bir bilgi yazıp kaydedin.
- ① **dir belge.txt** komutyla dosyanın boyutunu öğrenin.
- ① **%windir%\system32\notepad.exe belge.txt:gizlibelge.txt** komutuyla belge.txt içerisinde gizlibelge.txt dosyası oluşturup dosyanın içerisine gizli belgeyi kaydedin.
- ① **dir** komutuyla klasörde gizlibelge.txt dosyasının oluşturulup oluşturulmadığını ve belge.txt dosyasının boyutunda değişiklik olup olmadığını kontrol edin.
- ① Klasör seçeneklerinden gizli dosyaları ve sistem korunan dosyalarını görünür hale getirin ve gizlibelge.txt dosyasının görünürlüğünü kontrol edin.
- ① **%windir%\system32\notepad.exe belge.txt:gizlibelge.txt** komutuyla gizli belgenin içeriğini gözlemleyin.



# Uygulama - ADS

---

① **%windir%\system32\notepad.exe normal.txt**

① **type c:\resim.jpg>normal.txt:resim.jpg**

① **start .\normal.txt:resim.jpg**

① Normal dosya silindiğinde gizli dosyada silinir.

① Normal dosyayı silmeden gizli dosyayı silmek için aşağıdaki komutlar kullanılır.

① **ren belge.txt temp.txt**

① **type temp.txt > belge.txt**

① **del temp.txt**



# Uygulama - ADS

---

- ① **echo gizli bilgi > :hidden.dat** komutu kullanılarak herhangi bir metin klasöre gizlenebilir.
- ① Görüntülemek için **more < :hidden.dat** komutu kullanılır.
- ① Gizli belgeyi ayrı bir dosyaya ayırmak için.
- ① **more < belge.txt:gizlibelge.txt > gizli.txt** komutu kullanabiliriz.
- ① ADS kontrolü için 3part uygulamalar kullanılabilir : [AdsCheck](#), [Lads](#), [LNS](#), [Ads Spy](#), [Sfind](#), [Streams](#), [Hijackthis](#), [GMER](#), [Visual ADS Detector](#)



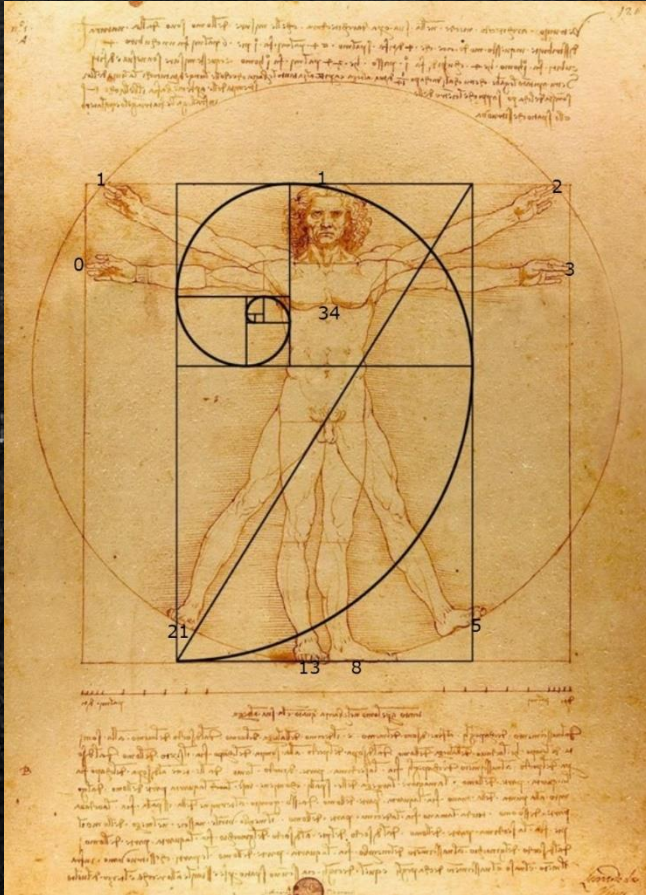
# Resimde Veri Gizleme Yöntemleri

---

- ⌚ Resim içinde mantıksal veri gizleme
- ⌚ Resim içine veri ekleme
- ⌚ Metadata alanına veri gizleme
- ⌚ Parlaklık değeri üzerinden veri gizleme
- ⌚ En önemsiz bite ekleme
- ⌚ Maskeleye ve filtreleme
- ⌚ Diğer algoritmalar ve dönüşümler.



# Resim İçinde Mantıksal Metin





# Resim sonuna veri ekleme

---

- ⌚ Veri resmin yada dokümanın sonuna data olarak eklenir.
- ⌚ Herhangi bir metin editöre vb. ile kolayca tespit edilebilir.
- ⌚ JPEG dosyasının başlangıç (0xFFD8) ve bitiş değeri (0xFFD9) değerleri önemli ayırt edici özelliktir.
- ⌚ Bu değerlere göre veri eklemesi yapıldığı ve iki verinin birbirinden ayrılabilir.



# Uygulama -

---

⌚ *Windows ortamında :*

⌚ *copy "<image file path>" + "<text file path>" "<new image path>"*

⌚ *Linux ortamında :*

⌚ *echo gizli bilgi >> resim.jpg* komutunu kullanarak bilgiyi resim dosyasına ekleyin.

⌚ *strings resim.jpg* komutunu kullanarak gizli bilgiyi resim dosyasından okuyun.

⌚ *Resim.jpg* dosyasını metin editörü ile açıp gizli bilgiyi elde edin.

⌚ *xxd resim.jpg* komutuyla resim dosyasının hex kodunu ve gizli bilgiyi görüntüleyin.



## Uygulama -

---

- ⌚ *Veri gizleme işlemini orijinal dosyayı saklı tutarak yapın ve eklediğiniz dosya boyutu ve eklenen dosya boyutundaki değişimi gözlemleyin.*
- ⌚ ***diff orijinal.jpg clone.jpg** komutuyla dosyada değışiklik olup olmadığını gözlemleyin.*
- ⌚ ***xxd orijinal.jpg orijinal.hex, xxd clone.jpg clone.hex** komutlarıyla dosyaların hex kodlarını kaydedin ve **diff orijinal.hex clone.hex** komutuyla iki dosya arasındaki farkı gözlemleyin.*
- ⌚ *Veri gizleme işlemini farklı formattaki dokümanlar için tekrar edin. Resim içerisine mp3, zip formatında veriler ekleyin. Dosyanın görünür kalıp kalmadığını kontrol edin.*
- ⌚ *JPEG dosyasının başlangıç (0xFFD8) ve bitiş değeri (0xFFD9) özelliğini kullanarak eklenen dosyayı jpg dosyasından ayırın.*



## Ödev -

---

- ⌚ Resim dosyaları için başlangıç ve bitiş hex değeri özelliklerini kullanarak resim dosyasına ekleme yapılıp yapılmadığını tespit eden program yazın.
- ⌚ Eğer veri eklenmiş ise eklenen veri türünü belirttin.
- ⌚ Eklenen dosyayı ayrı bir dosya olarak kayıt edebilme özelliği ekleyin.



# Metadata Alanına Veri Gizleme

---

- ⌚ Yöntemlerin en kolayı.
- ⌚ Veri, resim dosyasının metadata alanında gizlenir.
- ⌚ İdeal bir yöntem değil; metadata alanının sınırı var.
  - â Örn : yorum alanı 255 byte.
- ⌚ Bu yöntemde değişik resim dosya tipleri için değişik algoritmalar kurmanız gerekecektir.
- ⌚ Kolayca tespit edilebilir.
- ⌚ Resim dosyasına ait bir çok veri elde edilebilir. GPS lokasyonu, makine markası, tarih vb. Bu bilgiler güvenilir değildir, değiştirilebilir formattadır.



# Uygulama – Resim Açıklama Alanına Gizleme

---

- ⌚ **exiftool resim.jpg**
- ⌚ **exiftool –Comment=«eklenen metin» resim.jpg**
- ⌚ **exiftool boya.jpg** komutunu kullanarak gps lokaysounu, kullanılan kamera, tarih vb. bilgi edinin.
- ⌚ *Longitude ve latitude değerlerini kullanarak adresinden veya farklı bir araç kullanarak gps lokasyonun nereye ait olduğunu araştırın.*
- ⌚ **exiftool -GPSLongitude="14.273586" -GPSLatitude="50.860361" boya.jpg** komutunu kullanarak boya.jpg dosyasına ait lokasyon bilgisini değiştirin.



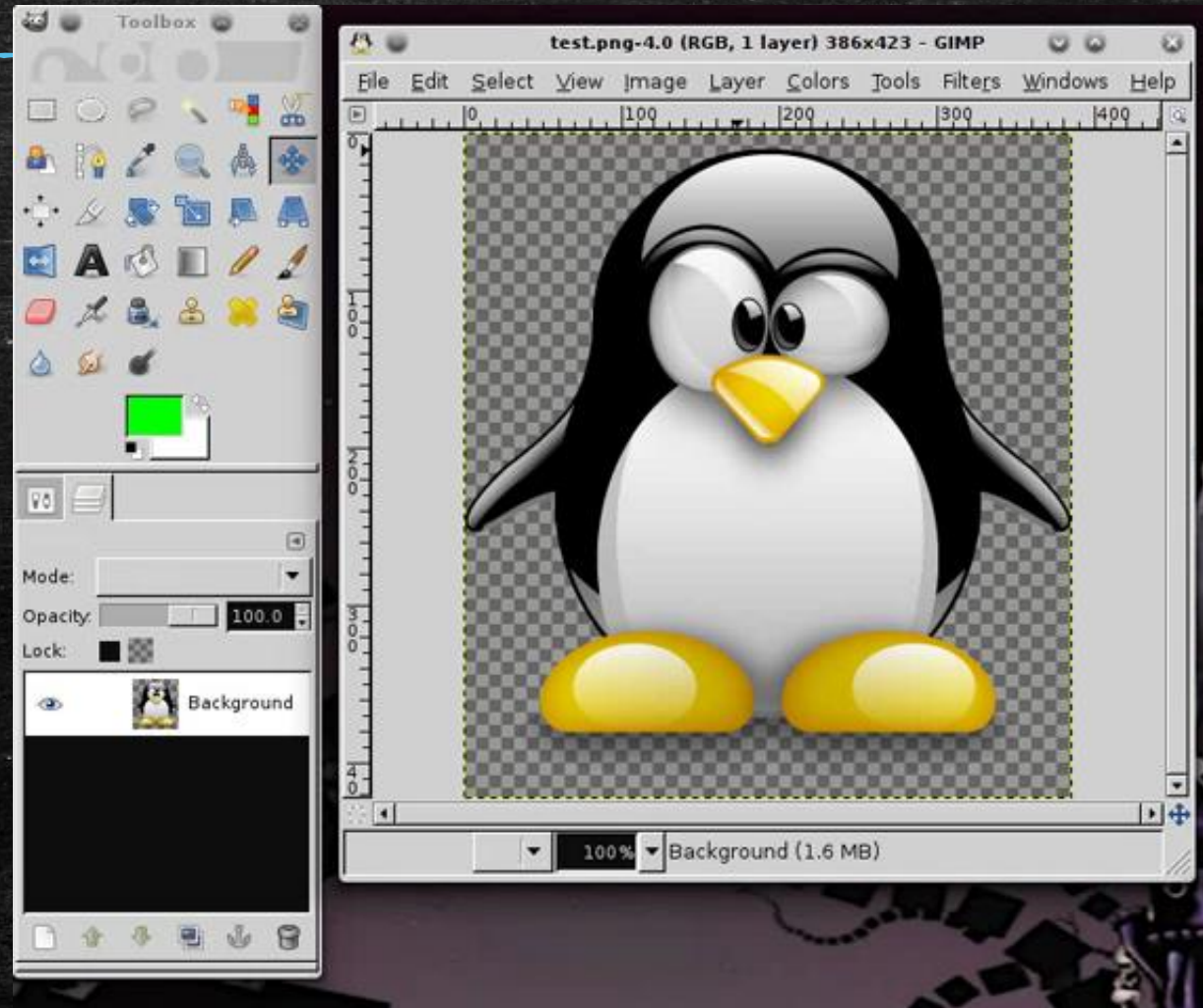
# Parlaklık Üzerinden Veri Gizleme

---

- ① Open your image in GIMP.
- ① Click *Windows > Dockable Dialogs > Palettes*. The image quality should degrade.
- ① Add a new Palette by clicking the blank document icon on the lower-left.
- ① Add 8 new colors by right clicking and bringing them from the foreground.
- ① Colors in order: ff0000:00ff00:0000ff:ffff00:ff00ff:00ffff:ffffff:000000
- ① Click *Image > Mode > Indexed*. Use our palette.
- ① Click *Windows > Dockable Dialogs > Colormap*.
- ① Change every color's FF to 02. The image should be almost black.
- ① Now click *Image > Mode > RGB*.
- ① Open a second image, and paste the first image we made into it.
- ① On the GIMP toolbar, anchor the image and select *Difference* from the mode menu.
- ① Click on *Image > Flatten Image*.
- ① Save as an uncompressed PNG file.



# GIMP ile Veri Gizleme





# Parlaklık Deęeri Üzerinden Veri Gizleme



- ⌚ Metin resim üzerine kaydedilir ve parlaklık değeri deęiştirilerek görünmez hale getirilir.
- ⌚ Yazılı metinlerin gizlenmesi için uygundur.
- ⌚ Genelde arka planı beyaz ve siyah olan resimler tercih edilir.
- ⌚ Küçük boyutlu veri gizleme için uygundur.
- ⌚ Resmin boyutunda büyük deęişiklikler olmaz.



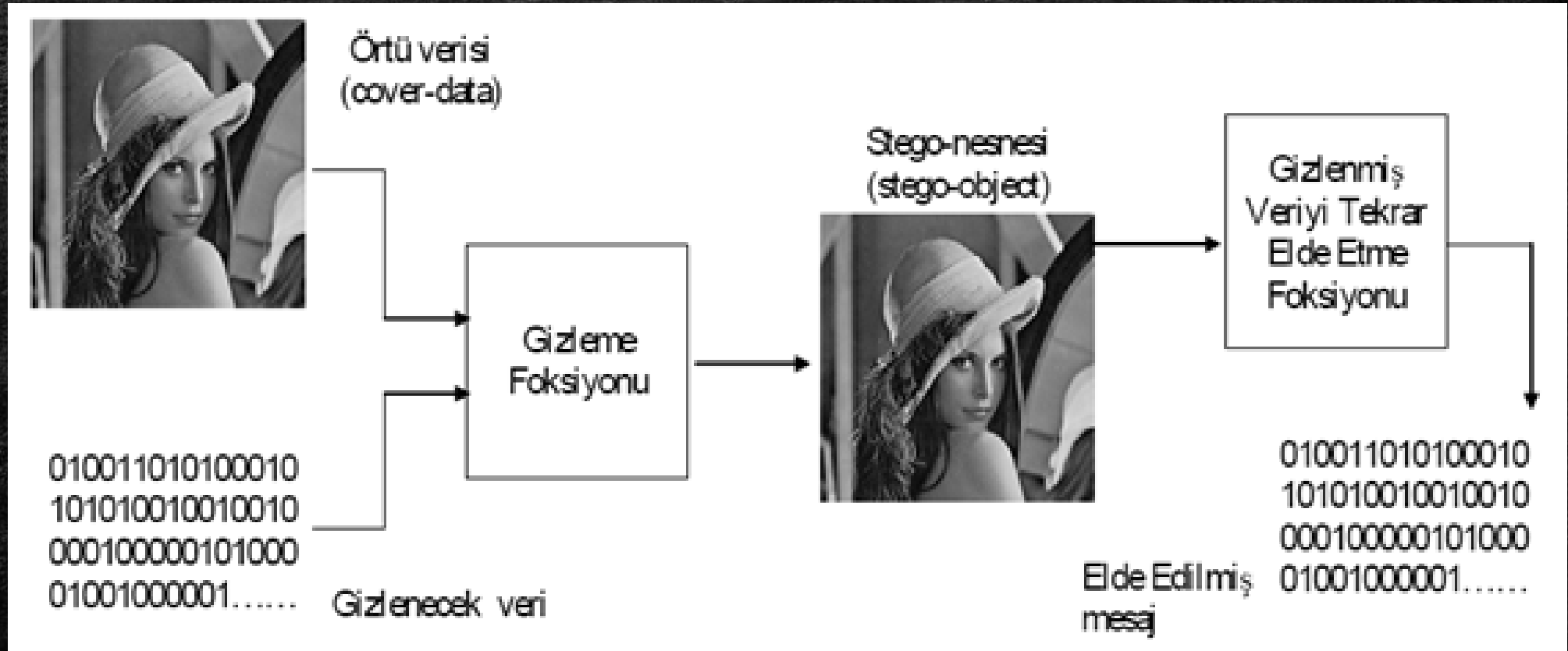
# Uygulama – Parlaklık Değeri İle Gizleme

---

- ⌚ Resim içinde gizlenmiş veriyi parlaklık değerini değiştirerek bulun.
- ⌚ Resim içerisine parlaklık değerini değiştirerek veri gizleyin. Bunun için basit olarak paint kullanabilirsiniz. Daha iyi çalışmalar için gimp resim editörünü kullanarak katmanlar oluşturabilirsiniz.
- ⌚ Fotomakron.jpg dosyası içerisinde gizli bilgiyi bulun.

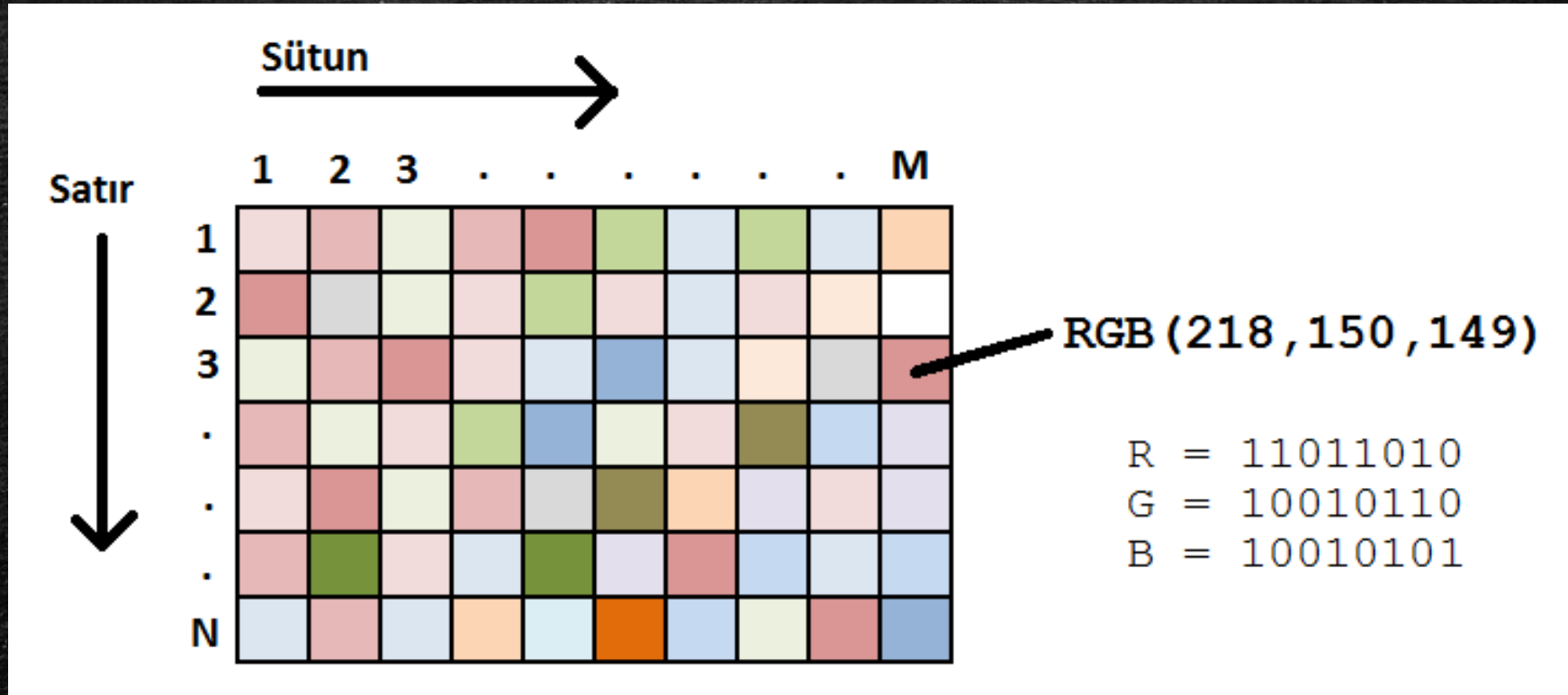


# Resimde Veri Gizleme





# Sayısal Resmin Yapısı



RGB (255, 127, 39)

RGB (255, 126, 39)



# Sayısal Resmin Yapısı

---

- ⌚ Bütün resimler dosya başlığı (header) ve piksellerden oluşur.
- ⌚ Sayısal (dijital) resim N satır ve M sütunluk bir dizi ile temsil edilir.
- ⌚ Bir resim dizisinin elemanlarına piksel denir.
- ⌚ Her piksel sadece bir renk içeren/gösteren küçük bir bloktur.
- ⌚ Bütün renk değişimleri üç temel rengin birleştirilmesiyle elde edilir. Basit dijital resim dosyasını ele alırsak, her temel renk 1 bayt ile gösterilir ve 24 bitlik bir resimde her piksel için renk değerini taşıyan 3 byte kullanılır.



# Resim Dosyalarında Renklerin Gösterimi

---

- ⌚ Günümüzde, analog bir imaj dosyasını dijital formata çevirdiğimiz zaman, renkleri sunmak için genelde üç farklı yol seçebiliriz:
- ⌚ **1 – 24 – bit color:** Her piksel  $2^{24}$  renkten birine sahip olabilir ve bunlar her 8 bit (256 değer) tarafından verilen, üç temel rengin ( R (red), gren (G), blue (B) ) farklı miktarları olarak sunulur.
- ⌚ **8 – bit color:** Her piksel  $2^8$  (256) renkten birine sahip olabilir ve renklerin bir tablosundan veya bir paletten seçilirler.
- ⌚ **8 – bit gray scale ( 8 bit gri seviye ):** Her piksel  $2^8$  (256) gri noktanın birine sahip olmalıdır.



# Örnek Resimler

---

⌚ 8, 24, gray scale resim örnekleri



# Resim Dosyalarında Renklerin Gösterimi

---

⌚ Her pikseldeki renk temel 3 rengin .(Red,Green,Blue) karışımından elde edilir. Her pikselde bu 3 rengin verileri tutulur. Her temel renk 1 pikselde 1 byte (0..255) yer kaplar, yani 1 piksel 3 byte (R,G,B) veri taşır.

⌚ **Örnek :**

Beyaz Renk RGB formatında FF-FF-FF hex değeri ile ifade edilir.

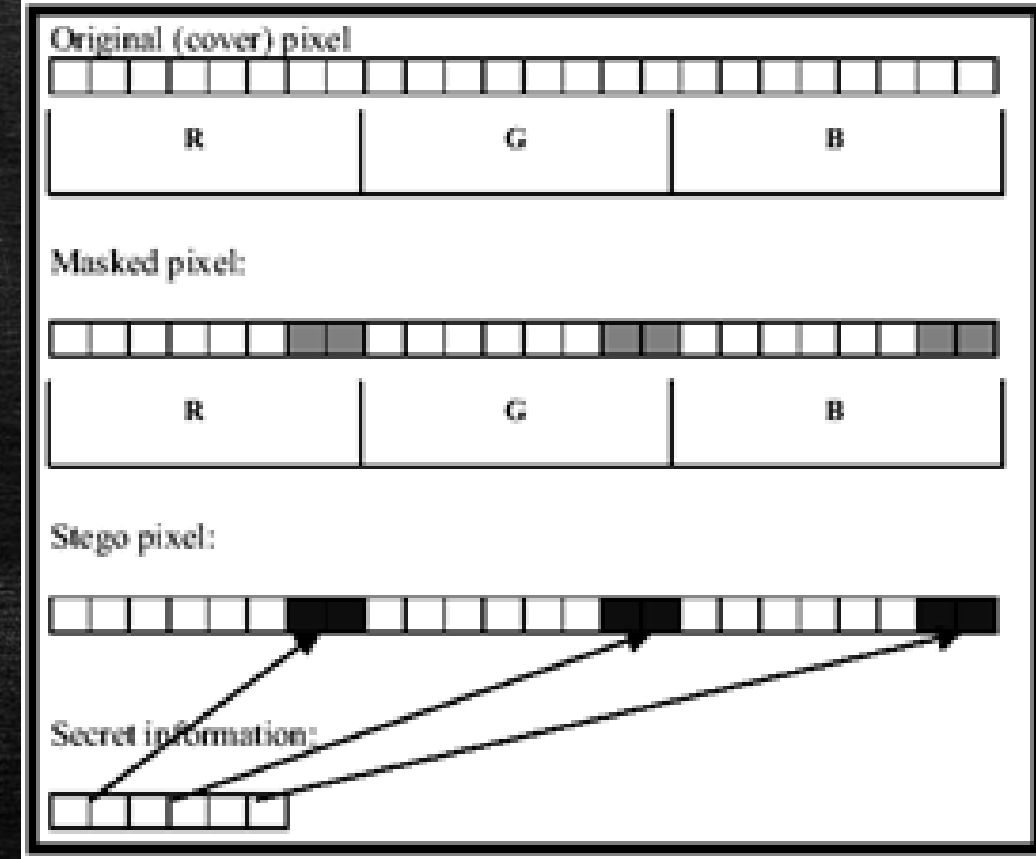
Yani Red: FF(h), Green: FF(h), Blue: FF(h)  
FF binary sistemde 11111111 in karşılığıdır.

1 pikselde beyaz rengi ifade etmek için 11111111- 11111111- 11111111 verisi kullanılır.



# Önemsiz bite ekleme(Least Significant (LSB))

- ⌚ LSB insertion yönteminde genel olarak son bitlere ekleme yapılır. Bununla birlikte son bitinde değiştirme yapmak yerine son 2 veya daha fazla bitinde değiştirme yapan yöntemlerde vardır.
- ⌚ İçine veri gömülen resimdeki değişim %50 oranındadır.
- ⌚ Bu durum insan gözüyle algılanamayacak oranda bir değişimdir.
- ⌚ Ne kadar fazla bit üzerinde değişiklik yapılırsa resim üzerindeki değişim o oranda artacağından algılanması da artacaktır.





# Önemsiz bite ekleme(Least Significant (LSB))

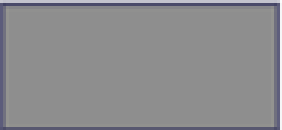
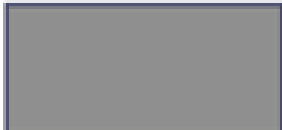
---

- ⌚ 24 bit renkli resimlerde renk spektrumu geniş olduğu için son 2 veya 3 bitteki değişim insan gözüyle algılanmayacak durumdadır.
- ⌚ Veri gömme işleminde ilk bayttan son bayta doğru bitlerin değişimi için bir sıra izleme zorunluluğu yoktur.
- ⌚ Verinin gömülmesi işleminde baytların hangi sırayla kullanılacağı genellikle “stego-key” anahtar vasıtasıyla düzenlenmekte ve tamamen karışık sırayla veri gömülmesi yapılabilmektedir.
- ⌚ Bazı akıllı yazılımlar koyu renkli alanları tespit edebilir ve bu alanların değiştirilmesini engeller.
- ⌚ Çünkü insan gözüyle koyu bölgelerdeki renk değişimlerinin algılanması daha kolaydır.



# Gri-seviye Resimlerde LSB yöntemi

- ❶ Gri-seviye resimlerde her piksel, 0 (siyah) ile 255 (beyaz) arasında tam sayı değer alabilen 1 byte ile temsil edilmektedir.
- ❷ 0–255 arasındaki değerler gri'dir ve bundan dolayı bir resme ait tam sayı "gri ton seviye" (gray level) olarak isimlendirilmektedir.
- ❸ Örneğin, renk değeri 142 olan bir pikselin içine ikilik sayı sistemindeki 1 değeri saklandığında oluşan piksel ve renk değeri aşağıda gösterilmektedir.

	Renk değeri	İkiliik Sistemdeki Karşılığı	Rengi
Orijinal piksel	142	10001110	
Bilgi saklanmış piksel	143	10001111	

**Son bitin 1 ya da 0 olması gözle görülebilir bir fark yaratmamaktadır.**



# 24-bit Renkli Resimlerde LSB yöntemi

- ① 24 bit resimler bir piksel başına 3 byte kullanmaktadır.
- ① Her pikselin rengi "Kırmızı (red), Yeşil (green), Mavi (blue)" olmak üzere üç ana renkten elde edilmektedir.
- ① Buna pikselin RGB değeri denmektedir.
- ① Her byte'ta son biti değiştirmek suretiyle bir pikselde 3 bitlik bilgi saklanabilir.
- ① Yani 24 bit derinliğine sahip  $1024 \times 768$  piksel boyutundaki 2.359.296 bit (294.912 byte) bilgi saklanabilir.

10010101 00001101 11001001 (149,13,201)

10010110 00001111 11001010 (150,15,202)

10011111 00010000 11001011 (159,16,234)

"101101101" bilgisi gizlendiğinde;

10010101 0000110**0** 11001001 (149,12,201)

1001011**1** 0000111**0** 1100101**1** (151,14,203)

10011111 00010000 11001011 (159,16,234)



# LSB ile veri gizleme

---

🕒 <https://www.mertsarica.com/pi-hediyem-var-di-verdim-gitti-3/>



# Uygulama -

---

- ⌚ stegdetect resim.jpg
- ⌚ steghide embed -cf icineGizlemeYapilacakDosya -ef gizlenecekDosya
- ⌚ steghide extract -sf [FILENAME]
- ⌚ steghide info [FILENAME]
- ⌚ stegbreak sözlük atağı
- ⌚ stegexpose
- ⌚ Hatıra resmi stegdetect ile gizleyiciyi bul
- ⌚ Basit bir wordlist ile steghide ile çöz



# Uygulama – EzStego ile Gif Üzerinde Gizleme

---

- ⌚ EzStego gif imaj üzerinde daha karmaşık şifrelemeler yapılabilir.



# Ödev -

---

- ⌚ Şifre anahtar değerine göre LSB ile resim içerisine veri gizleyen program yazın.
- ⌚ Anahtar değeri çift veya tek bite şifreleme yapacağını belirtecek.



# Ses İçerisine Veri Gizleme

---

- ⌚ Ses melodisinde bozukluğa neden olur.
- ⌚ Anlaşılması daha kolaydır.
- ⌚ Klasik müziklerde daha kolay anlaşılır.
- ⌚ Gizleme için rock ve bir çok enstrümanın kullanıldığı çok sesli müzik türleri daha uygundur.



# Ses İçerisine Veri Gizleme Yöntemleri

---

⌚ LSB CODING

⌚ PARITY CODING

⌚ PHASE CODING

⌚ SPREAD SPECTRUM

⌚ ECHO HIDING

⌚ <https://iicybersecurity.wordpress.com/2015/08/25/how-to-hide-secret-messages-in-music-files/>

⌚ <http://www.snotmonkey.com/work/school/405/methods.html>

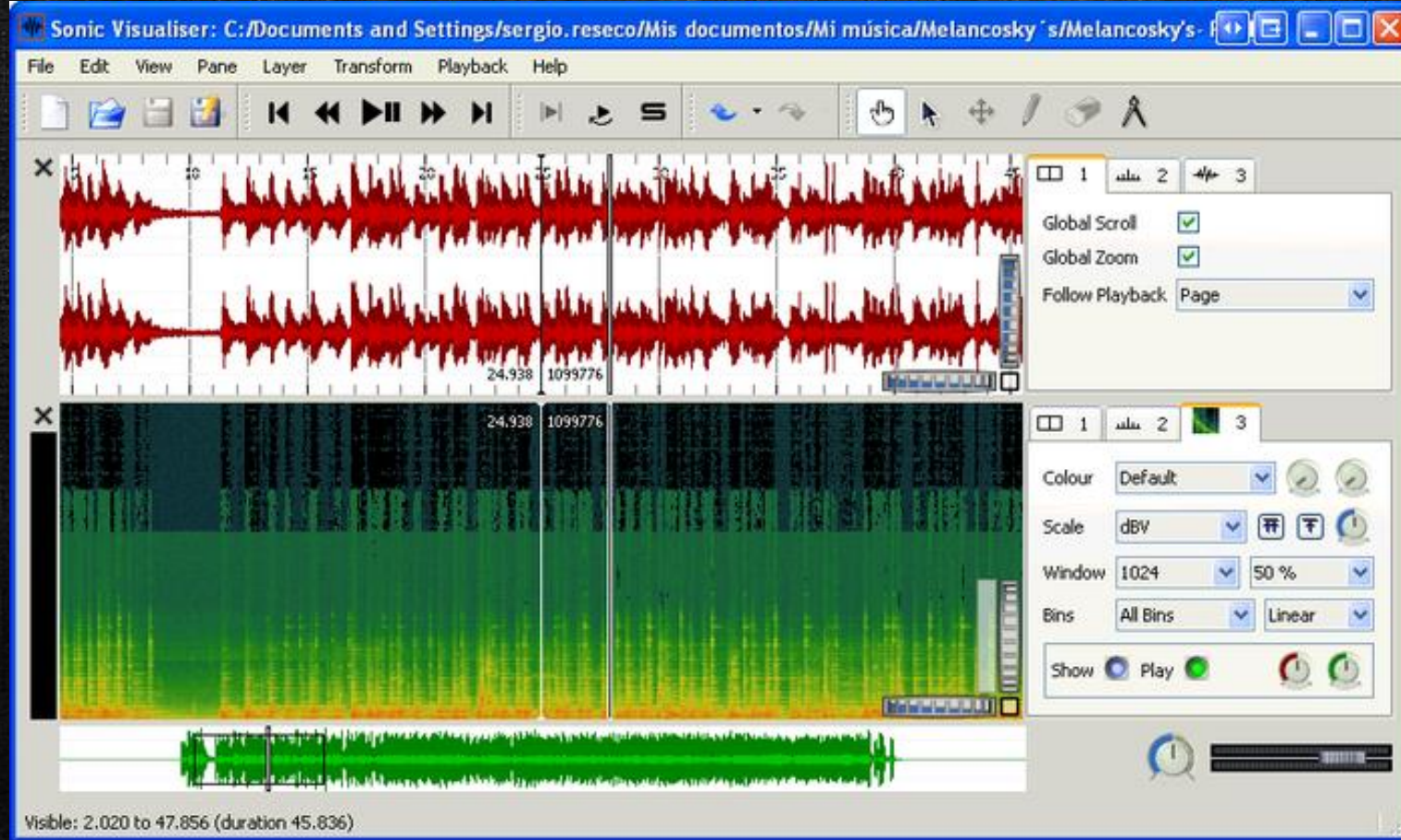


# Uygulama – Morse Şifreleme

---



# Uygulama – Spread Spectrum



- ⌚ mix.mp3 müzik dosyasını dinleyip gizlenen bilgi olup olmadığı kontrol edin.
- ⌚ mix.mp3 dosyasını [sonic visualizer](#) programında açın.
- ⌚ Programın layer bölümünden add melodic range spectrogram seçip kanallar ile oynayarak gizli bilgiyi bulun.
- ⌚ World on fire.wav dosyası için benzer çalışmayı yapın.



# Spread Spectrum Frekans İle Veri Gizleme

---





# Uygulama - Spectrum Frekans İle Veri Gizleme

---

- ⌚ Şifrelenmesi istenen resmi oluşturun.
- ⌚ Siyah arkaplan üzerine beyaz metin daha iyi sonuç veriyor.
- ⌚ Fazla boş alan olmayan küçük resimler daha iyi sonuç veriyor.
- ⌚ Oluşturduğunuz resmi jpeg veya bmp formatında kaydedin.
- ⌚ Windows üzerinde çalışan [coagula](#) programını indirin.
- ⌚ Resmi Coagula programı ile açın. Ayarlar menüsünden "Render Image as Audio Without Blue/Noise" seçin.
- ⌚ Menüden "Save Audio File" seçerek ses dosyasını kaydedin.



# Uygulama – Müziksel Şifreleme

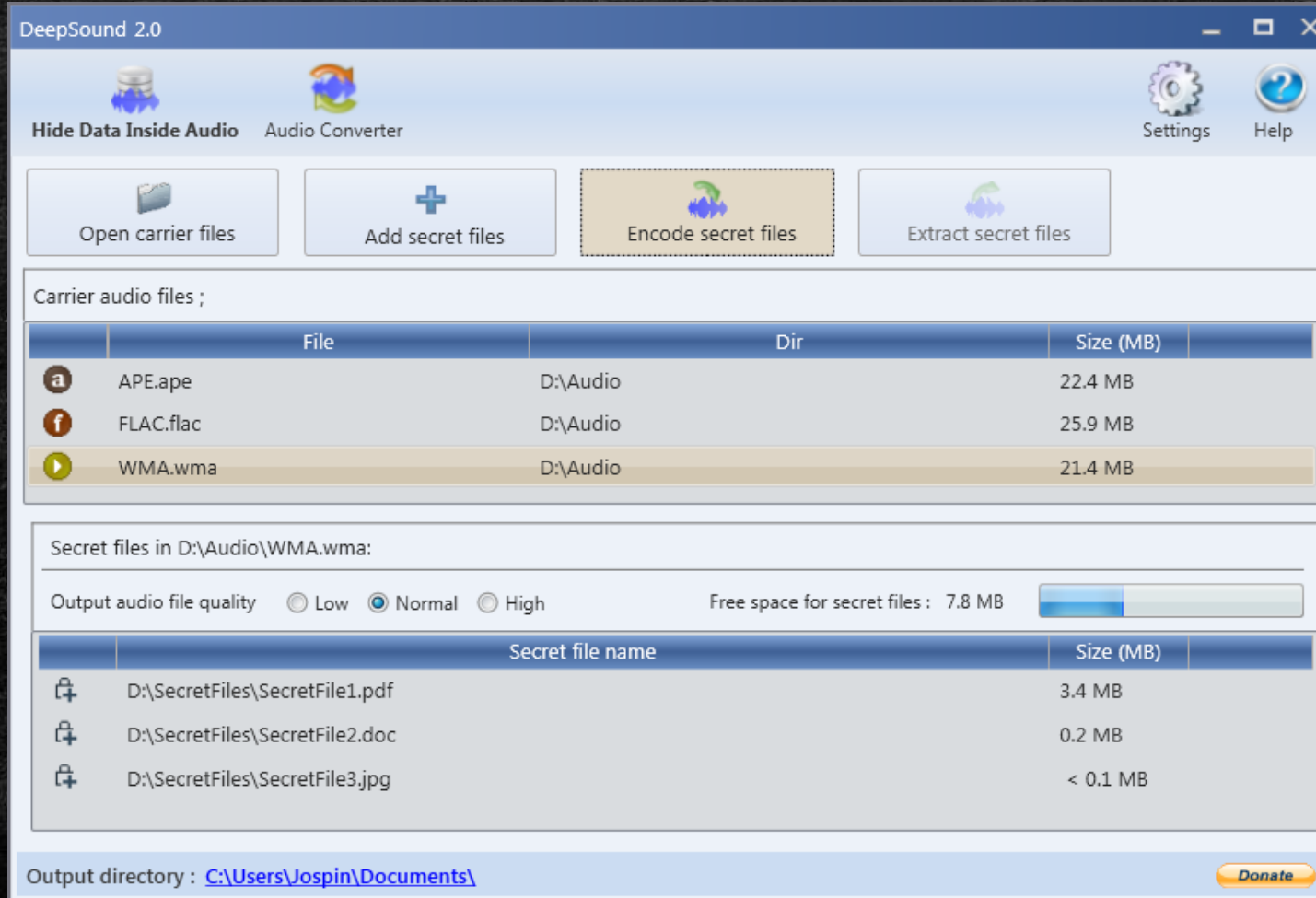
---

- ① <https://p22.com/musicfont/> web sayfasından gizlemek istediğiniz bilgiyi müzik dosyasına dönüştürün.
- ① <http://www.dcode.fr/music-sheet-cipher> web sayfası üzerinden gizlemek istediğiniz veriyi notalara çevirin. ([Musical Cryptogram](#))



# Uygulama – DeepSound ile Şifreleme

- ⌚ Windows üzerinde çalışan [deepSound](#) uygulamasını kurun.





# DTMF (Dual-Tone Multi-Frequency) kodları

---

- ⌚ Amerikan ordusu için Bell telefon laboratuvarlarında geliştirilmiş bir kodlama sistemidir.
- ⌚ Telefon şebekelerinde bilgi yollamanın güvenli yolu olarak tercih edilmiş ve telefon abonesinin santrale aradığı abone ile ilgili bilgileri ilettiği standart yöntem olarak günümüzde yaygın olarak kullanılmaktadır.
- ⌚ Günümüzde telsiz cihazlarının çoğunda DTMF sinyallerini yollamaya ve almaya yarayan modüller ile tuş takımı bulunmaktadır.
- ⌚ Bu sinyallerin çözülmesi oldukça kritik ve karmaşık bir DSP – Digital Signal Processing – tekniği gerektirmektedir.
- ⌚ Çözümleme için geçerli bir ton çifti ve zamanlama aralığı gereklidir.
- ⌚ Çözüm için genelde Goertzel algoritması kullanılıyor.



# DTMF (Dual-Tone Multi-Frequency) kodları

DTMF tuş takımı frekansları				
	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	<u>1</u>	<u>2</u>	3	<u>A</u>
770 Hz	<u>4</u>	5	<u>6</u>	<u>B</u>
852 Hz	<u>7</u>	<u>8</u>	9	<u>C</u>
941 Hz	<u>*</u>	<u>0</u>	<u>#</u>	<u>D</u>



# Uygulama – Telefon Tuş Seslerinin Tespiti

---

- ⌚ [Detect DTMF \(Dual Tone Multi Frequency\)](#) adresine DTMF\_dialing.wav ses dosyasını yükleyerek tuşlanan numaraları bulun.
- ⌚ Çağrı merkezi görüşmesi ses dosyası içerisindeki kredi kartı numarasını elde edin. (Kredi kartı numara kısımlarını ayrı ayrı elde ederek çalışmak sizi daha kolay sonuca ulaştıracaktır.)
- ⌚ Online web sayfaları yanı sıra [dtmf2num](#), [DTMFChecker](#), [ToneDecoder](#) programı da kullanılabilir.



# Uygulama – Ters Çevirilmiş Ses Dosyası

---

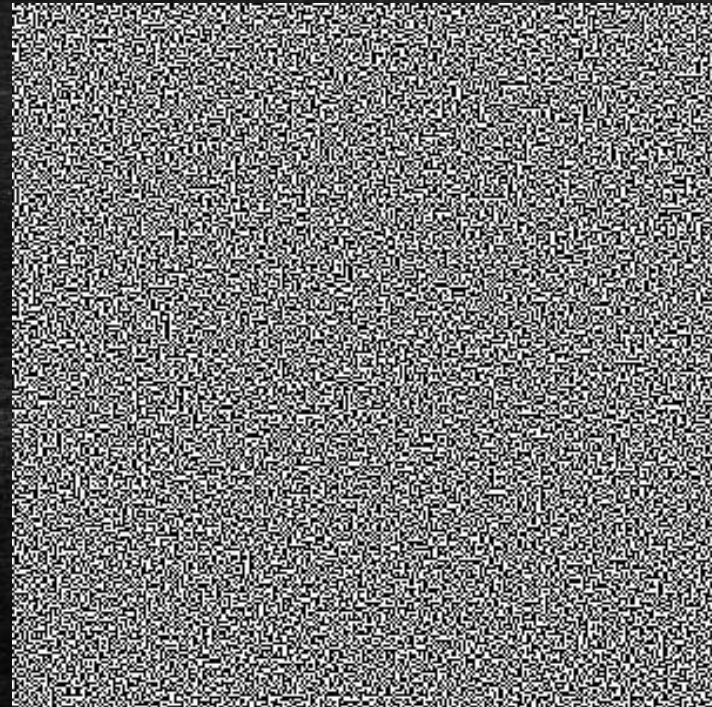
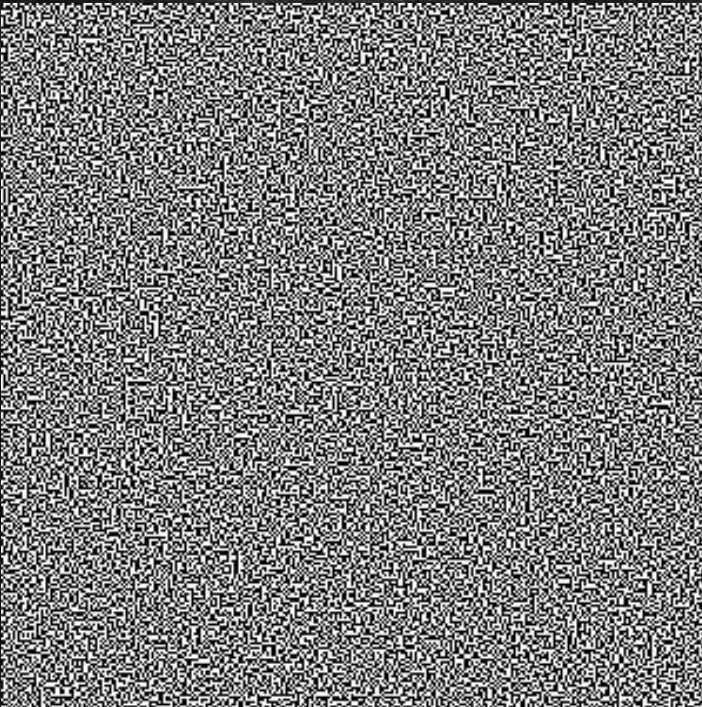
- ⌚ Auodacity programı ile reverse\_sound ses dosyasını ters çevirip dinleyin.



## Uygulama - XOR

---

Resimleri XOR'layıp ortak resmi bulun. [Stegsolver](#) aracını kullanabilirsiniz.





# Uygulama -



binwalk moon.png veya  
binwalk -Me firmware.bin  
dd if=./moon.png of=./moon.zip skip=411781 bs=1

- ⌚ xxd carving-ex.jpg | less
- ⌚ PK değeri zip dosya sistemi oluşturan [Phil Katz](#) ifade ettiğinden resmin sonunda zip dosyası olduğunu anlayabiliyoruz.
- ⌚ printf '%d\n' 0x00008e23
- ⌚ dd if=carving-ex.jpg bs=1 skip=36387 of=foo.zip
- ⌚ Resim dosyasından zip dosyasını ayırmak için binwalk programını kullanabilirsin.



## Uygulama -



Benzer işlemi yandaki  
resim için uygulayın



# Uygulama – Steg Solve

---





# Video İçine Veri Gizleme

---

⌚ Stego Video



# Kaynakça

---

1. İsmail Saygılı, [ADS Kullanarak Veri Gizleme ve Steganography Tekniği](#)
2. Sawat Pathi, [Hide Data WITHOUT Steganography](#)
3. Chetan Gupta, [Dissecting NTFS Hidden Streams](#)
4. Selda Berker, [Steganografi ve LSB](#)
5. Adam Dachis, [How to Hide Secret Messages and Codes in Audio Files](#)
6. Mantisreligiosa, [DTMF Ton kodlama sistemi](#)
7. <https://ctfs.github.io/resources/topics/steganography/file-in-image/README.html>
8. <https://ctfwriteups.wordpress.com/tag/steganography/>



# Kaynaklar

---

- ⌚ Alex Long, [A Guide to Steganography, Part 3: How to Hide Text and Images in Pictures](#)
- ⌚ Shankar Raman, [Archive for the 'Stegano' Category](#)
- ⌚ <https://github.com/ctfs/>



---

① <http://www.slideshare.net/nulloxoo/nullcon-2010-steganography-stegananalysis-a-technical-psychological-perspective>