**DİNAMİK S-BOX KULLANILARAK ŞİFRELENMİŞ RESMİN QR KODA DÖNÜŞTÜRÜLMESİ**

**Devran CANİKLİ**

**Eren Alp ŞAKACI**

**Karabük Üniversitesi**

**Bilgisayar Mühendisliği**

**Tez Danışmanı:**

**Dr. Öğr. Üyesi Burhan SELÇUK**

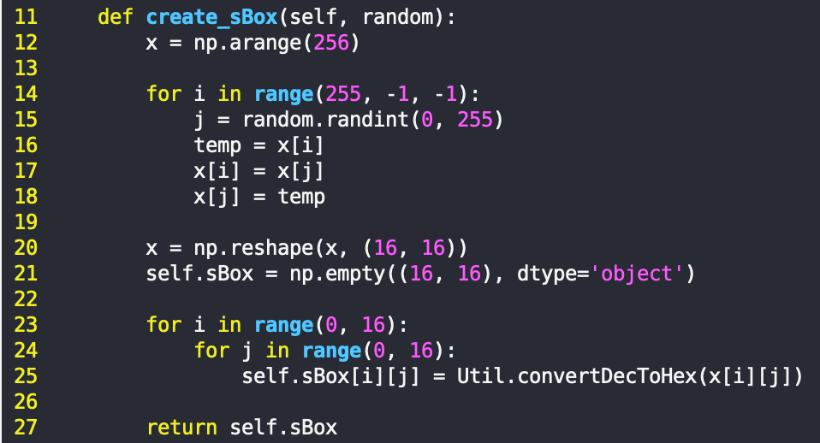
**Özet:** Günümüzde mesajlaşma uygulamalarının yaygınlaşması sebebiyle gün içerisinde yoğun bir şekilde resim alışverişi yapılmaktadır. Bu da çeşitli güvenlik ve gizlilik sorunlarına neden olmaktadır. Bu çalışmada hassas içeriğe sahip resimlerin bir anahtar ile şifrelenip, QR koda dönüştürülerek güvenliğinin ve gizliliğinin artırılması amaçlanmıştır. Şifrelenen resim sadece anahtarı bilen kullanıcı tarafından çözülebilmektedir. Bu sayede resmin paylaşıldığı ortama saldırılması halinde üçüncü şahıslar resmin içeriğini görüntüleyememektedir. Şifreleme algoritmasında dinamik S-box kullanılmıştır.

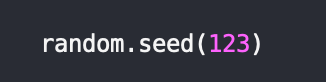
**Giriş**

AES (Advanced Encryption Standard) veya DES (Data Encryption Standard) gibi blok şifreleme algoritmalarının, resim dosyalarının boyutlarının büyük olmasından dolayı resim şifreleme için kullanılması uygun görülmemektedir [1]. Bu çalışmada AES’in temelinde bulunan S-box (substitution-box) yöntemi kullanılmıştır. S-box, simetrik şifreleme algoritmalarında yer değiştirme (substitution) amacıyla kullanılan bir tekniktir. Blok şifrelemede kullanılan S-box’lar genellikle statik bir şekilde üretilmektedir. Bu çalışmada kullanılan S-box güvenliği artırmak amacıyla belirlenen anahtara göre dinamik bir şekilde üretilmiştir.

**1. S-box Üretimi**

Çalışmada kullanılan S-box 0-255 aralığında 256 adet sayı içermektedir. Bu sayıları 16x16 boyutundaki matrise dinamik (rastgele) olarak yerleştirebilmek için öncelikle bir seed (anahtar) değeri belirlenmiştir. Bu anahtar Knuth Shuffle algoritmasında sayıları karıştırmak amacıyla kullanılmıştır [2].

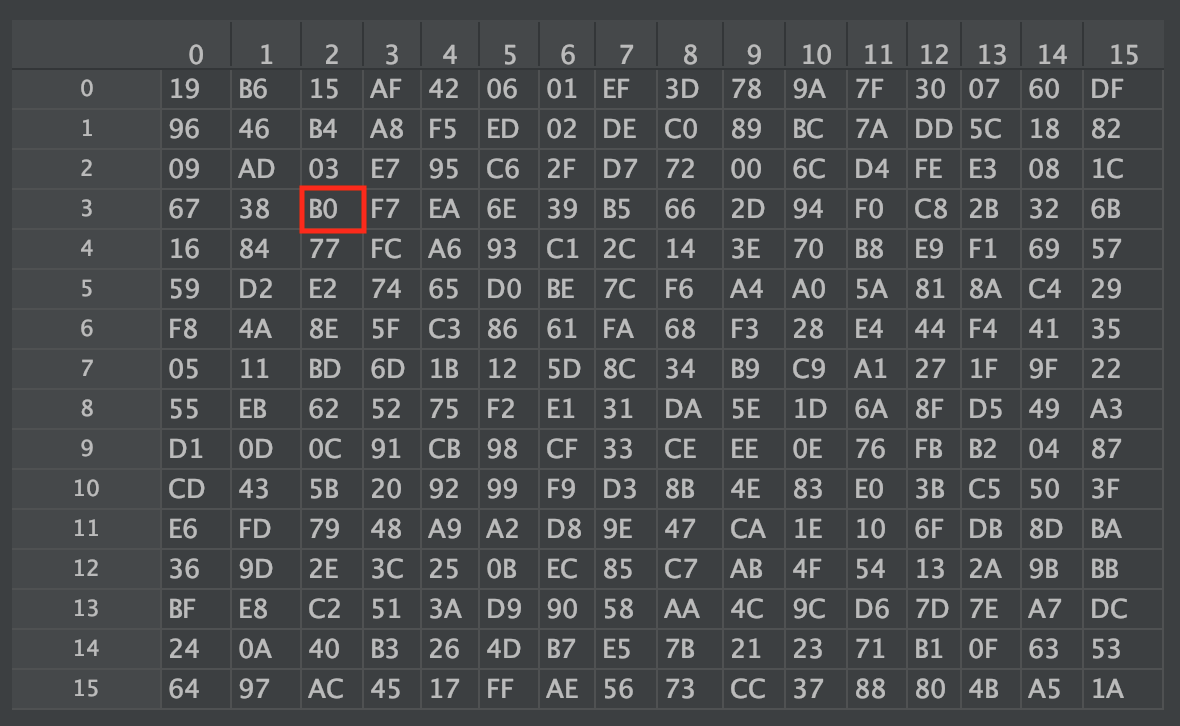




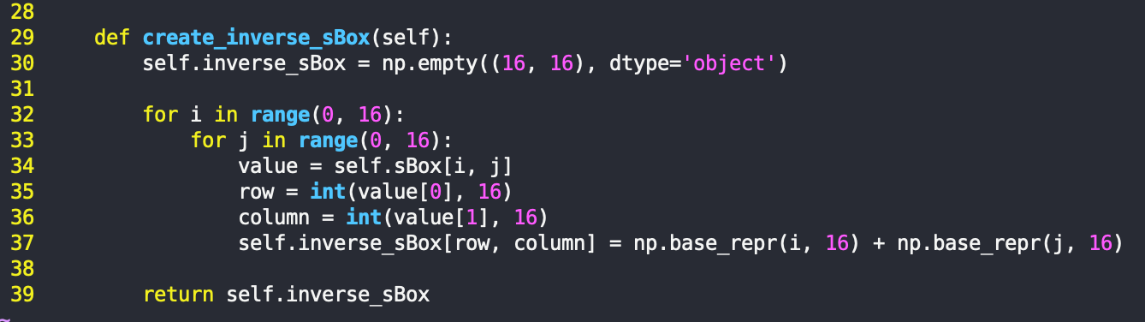
Şekil 1: Seed Değerinin Belirlenmesi

Şekil 2: S-Box’ın Üretilmesi.

Şekil 2’de bulunan **‘create\_sBox’** metodu 16x16 boyutunda 0-255 aralığında rastgele sıralanmış 256 adet sayılardan oluşan matris döndürmektedir. 12. Satırda 0-255 aralığında sıralı bir dizi oluşturulmaktadır. 14. satırda başlayan for döngüsü Knuth Shuffle algoritması ile dizi içerisinde bulunan sayıları karıştırmaktadır. Elde edilen tek boyutlu dizi 20. satırda 16x16 boyutunda matrise dönüştürülmektedir. 23. satırdaki for döngüsü, oluşturulan x matrisindeki değerleri on altı tabanındaki sayılara dönüştürerek S-Box matrisine atamaktadır.

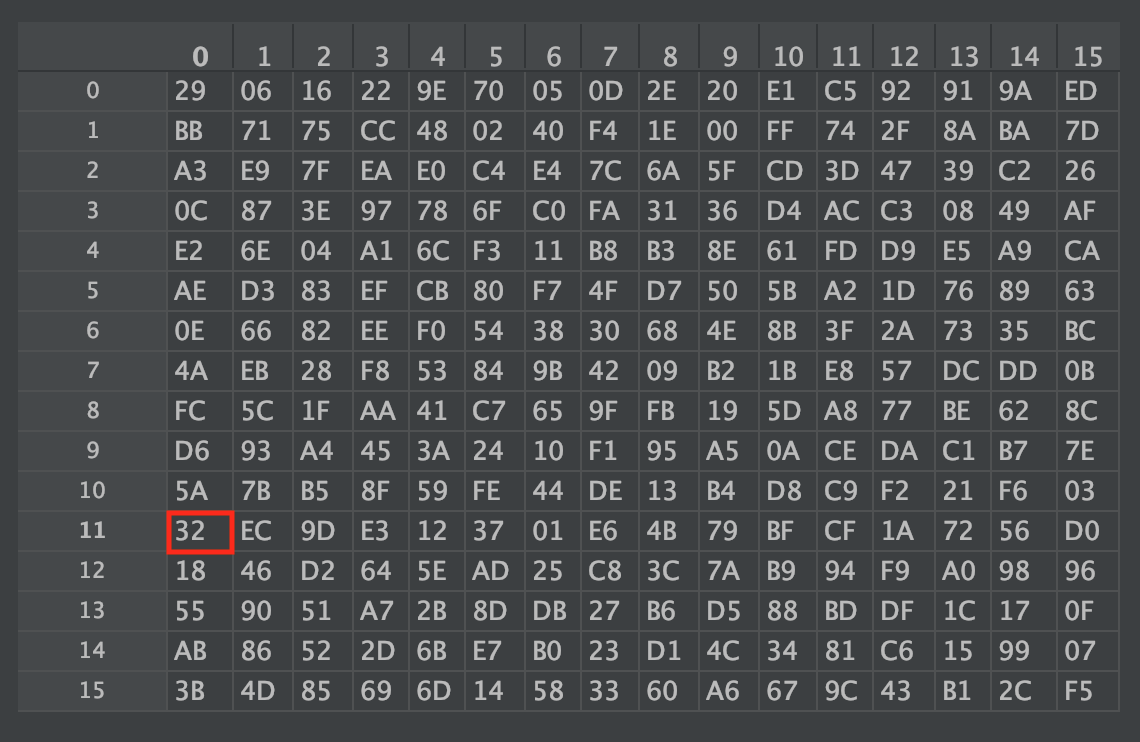


Şekil 3: ‘123’ Seed Değerine Göre Üretilen S-Box.

Deşifreleme aşamasında kullanılan ters (inverse) S-box’ın üretildiği metot aşağıdaki gibidir.

Şekil 4: Ters S-Box üretilmesi.

Ters S-Box, S-Box değerlerine bağlı üretilen bir matristir. Örneğin, S-Box matrisinde 3. Satır 2. Sütundaki değer B0’dır. Ters S-Box’ta B. Satır (11. Satır) 0. Sütundaki değer 32 olacaktır.



Şekil 5: ‘123’ Seed Değerine Göre Üretilen Ters S-Box.

**2. Resmin Şifrelenmesi**

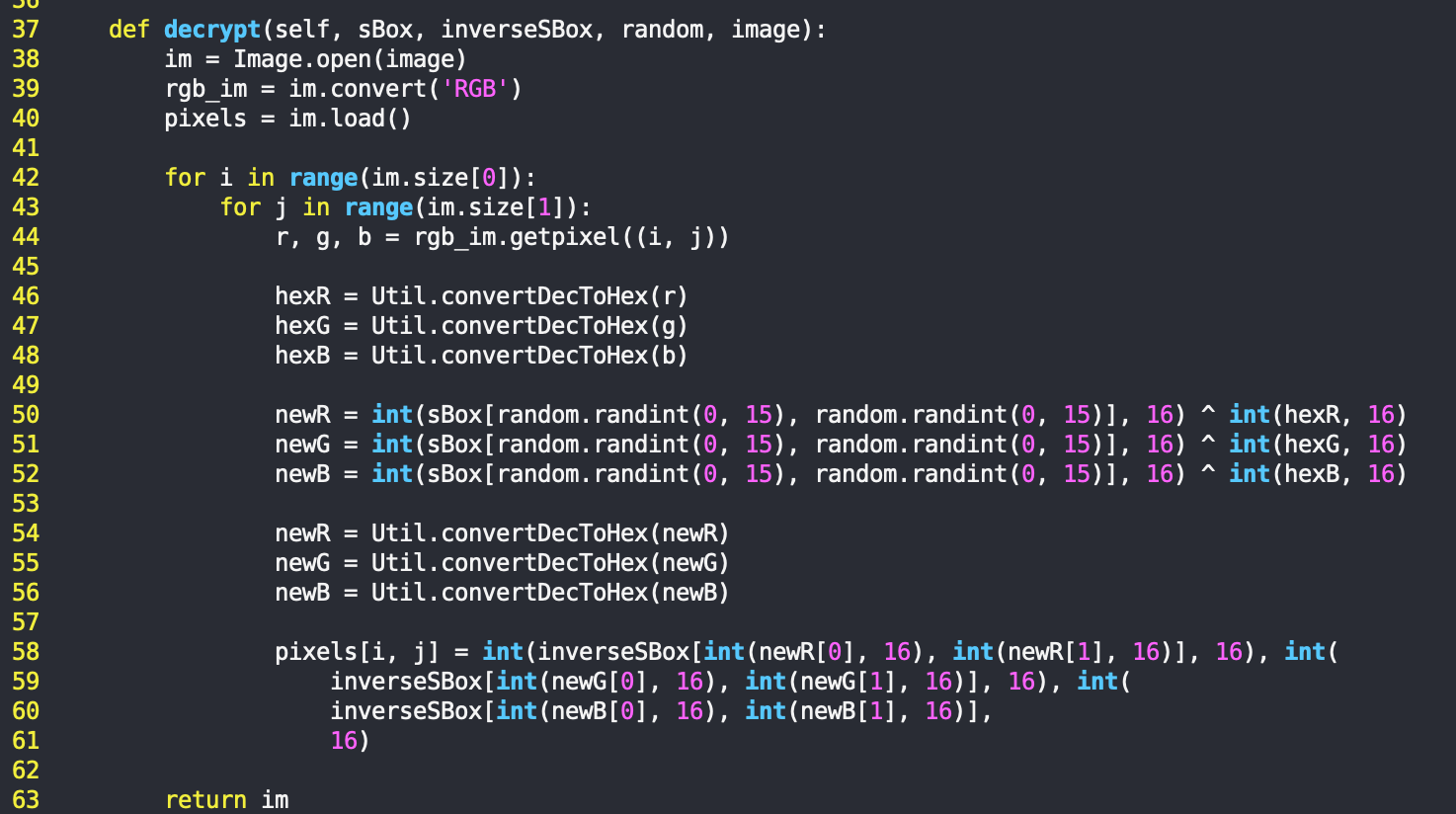
Şifreleme metodu aşağıdaki gibidir:



Şekil 6: Şifreleme Metodu

Şekil 6’daki şifreleme metodu parametre olarak üretilen S-box’ı, random nesnesini ve şifrelenmesi istenen resmi almaktadır. 8. Satırda okuna resim 9. Satırda ‘RGB’ formatına dönüştürülmektedir [3]. Bu değerler ‘pixels’ matrisine atanmaktadır. For döngülerinin içinde her bir pikselin R (Red), G (Green), B (Blue) değerleri 16 tabanında sayıya dönüştürülmektedir. Örneğin, R değerini E8 olarak kabul edelim. S-box ta kullanılmak üzere E değeri satır, 8 değeri sütun olarak atanır. 29, 30, 31. Satırlarda xor işlemi gerçekleşmektedir. S-box’ın rastgele satır ve sütunundaki değer ile S-box’ın E. Satır, 8. Sütunundaki değer xor işlemine tabi tutulur. İşlem sonucunda oluşan sayı yeni R değeri olarak belirlenir. Bu işlem bütün piksellerin RGB değerleri için tekrarlanır. Metot sonunda şifrelenmiş resim döndürülür.

**3. Resmin Deşifrelenmesi**

Deşifreleme metodu aşağıdaki gibidir:

Şekil 7: Deşifreleme Metodu

Şekil 7’deki deşifreleme metodunda, şifreleme metodundaki gibi resmin RGB değerleri okunur. 50, 51 ve 52. satırlarda S-box’ın rastgele satır ve sütun değerleri ile şifrelenmiş resmin RGB değerleri xor işlemine tabi tutulur. İşlem sonucunda elde edilen RGB değerleri orijinal resmin RGB değerleri ile aynıdır. Bu değerler ‘pixels’ matrisine atanır. Metodun sonunda deşifrelenmiş resim döndürülür.

İşlemler sonucunda elde edilen şifrelenmiş ve deşifrelenmiş resimler aşağıdaki gibidir:



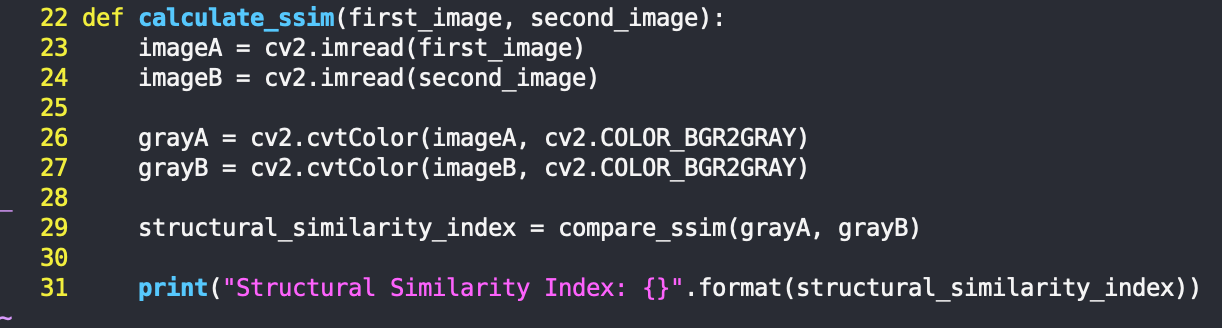
Şekil 10: Deşifrelenmiş Resim

Şekil 9: Şifrelenmiş Resim

Şekil 8: Orijinal Resim

**4. Yapısal Benzerlik İndeksi**

Orijinal resim ile deşifrelenmiş resmin yapısal benzerlik indeksi hesaplanarak indeks değerinin 1.0 olduğu gözlendi. Bu değer, orijinal resim ile deşifrelenmiş resmin birebir aynı olduğunu göstermektedir [4].

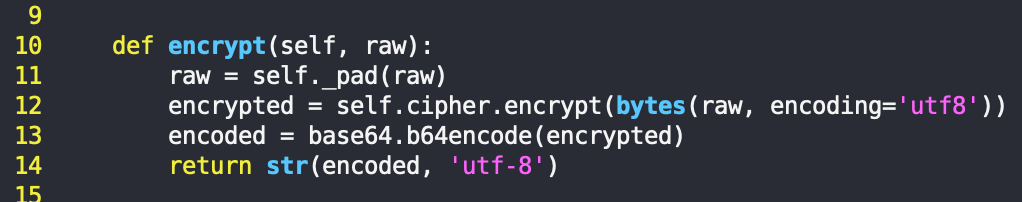


Şekil 11: Yapısal Benzerlik İndeksinin Hesaplanması.

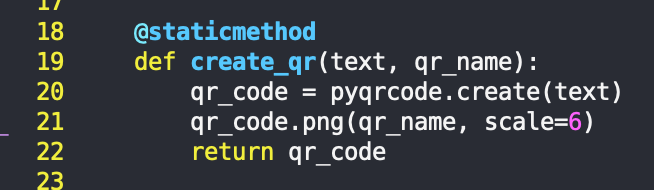


Şekil 12: Yapısal Benzerlik İndeksi.

**5. QR Kod Oluşturulması**

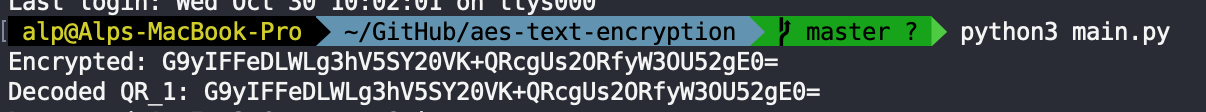
QR kodun kullanımını anlamak amacıyla bir metin AES ile şifrelenip QR kod içine gömüldü [5]. Metnin şifrelenme metodu aşağıdaki gibidir.

Şekil 13: AES şifreleme metodu.

Şekil 13’teki metot parametre olarak plain text almaktadır. Metin AES ile şifrelendikten sonra base64 encode şeması ile kodlanır ve şifrelenmiş metin döndürülür.

Şekil 14: QR kod oluşturma metodu.

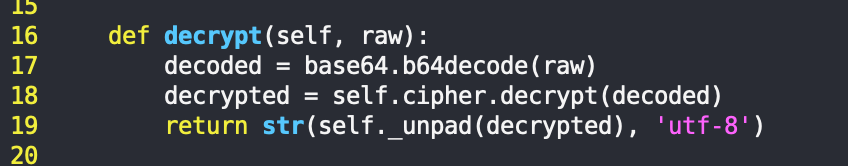
Python’un pyqrcode kütüphanesinde bulunan metot ile QR kod oluşturuldu [6]. Metot parametre olarak şifrelenmiş metni almaktadır.



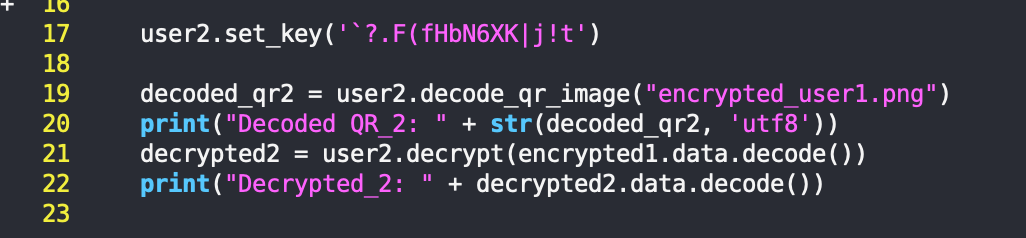
Şekil 15: Şifrelenmiş metin ve QR kodun içerdiği string.

Şekil 16: Oluşturulan QR kod.

QR kodun içerisinde bulunan veri alındı. Şifreleme aşamasında kullanılan anahtar ile deşifrelendi.



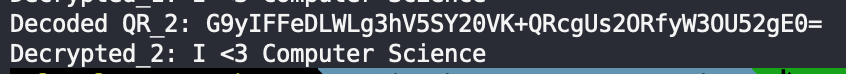
Şekil 18: Deşifreleme metodu.



Şekil 19: Şifreli metin bulunan QR kodun anahtar ile deşifrelenmesi.

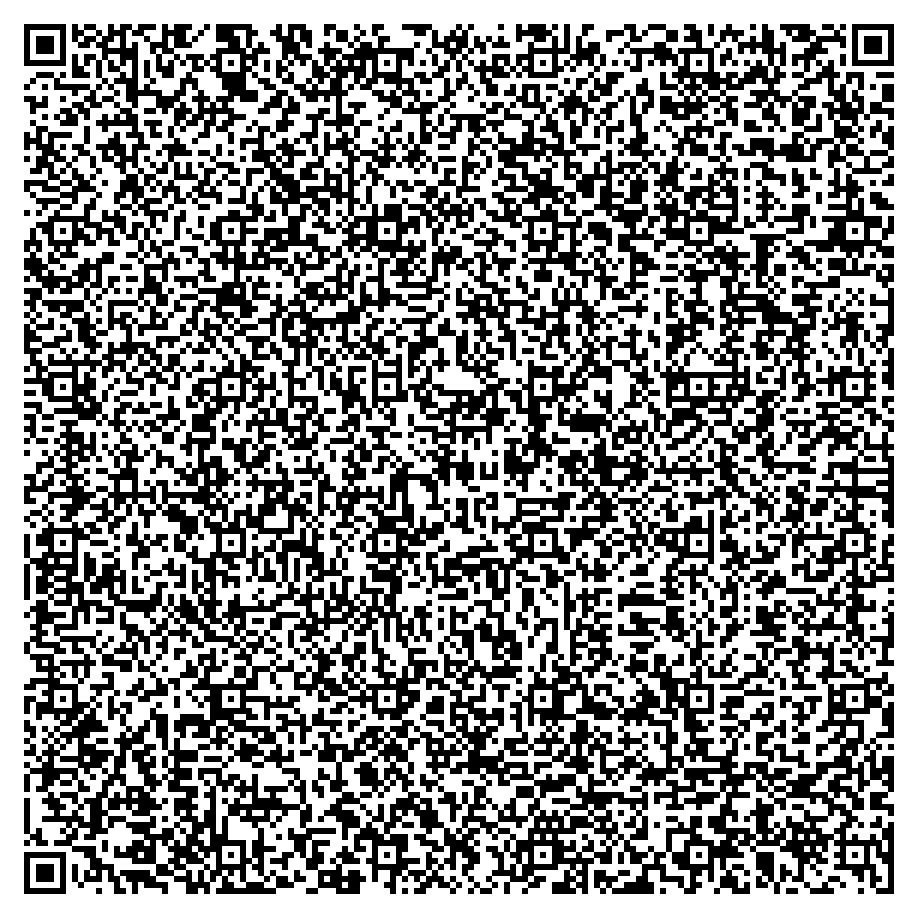
Şekil 17: QR kodun decode edilmesi.

Deşifreleme işlemi sonucunda elde edilen metin aşağıdaki gibidir. Şifrelenmek istenen metin ile deşifrelenmiş metnin aynı olduğu görüldü.



Şekil 20: Deşifreleme metodunun çıktısı.

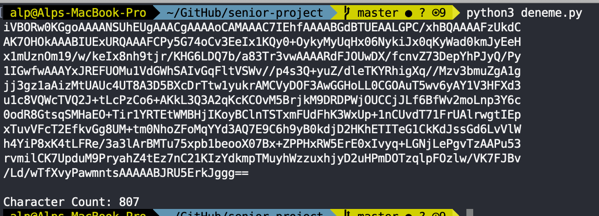
Yapılan çalışmaya ek olarak QR kod içerisine resim yerleştirildi. Görselin pikselleri base64 kodlama şeması ile string’e çevrilerek QR koda yerleştirildi.





Şekil 21: QR kod içine yerleştirilecek resim.

Şekil 22: Resmi içeren QR kod.



Şekil 23: QR kodun içerdiği string.

QR kodun içerdiği string base64 ile decode edildiğinde orijinal resme ulaşıldı.

Şekil 23: Elde edilen resim.

**Kaynaklar**

[1] Güvenoğlu, E., "Resim Şifreleme Amacıyla Dinamik S Kutusu Tasarımı İçin Bir Yöntem", El-Cezerî Fen ve Mühendislik Dergisi, 29 Şubat 2016

[2] Wikipedia. Fisher-Yates Shuffle – Wikipedia, the free encyclopedia

[3] Python Pillow Kütüphanesi Image Modülü, Pillow Documentation, https://pillow.readthedocs.io/en/stable/reference/Image.html

[4] Structural Similarity Index , Python Scikit-Image Documentation, https://scikit-image.org/docs/dev/auto\_examples/transform/plot\_ssim.html

[5] Pycrypto API Documentation, AES Module, https://www.dlitz.net/software/pycrypto/api/current/

[6] Python Pyqrcode Documentation, Creating QR Codes, https://pythonhosted.org/PyQRCode/create.html