

Güvenli Dosya Transferi İçin Şifreleme Tabanlı ağ Uygulaması

PROJE ARA RAPRORU



27 Nisan 2025

Bursa Teknik Üniversitesi

Yusuf Güney - 22360859041

**1. GIRIŞ**

Bu projede amacımız, ağ üzerinde güvenli şekilde dosya transferi gerçekleştiren, temel şifreleme, paket parçala/birleştir ve IP başlığı işleme gibi ağ seviyesi kavramları uygulayan bir sistem geliştirmektir. Ağ üzerinden gerçekleşen veri iletiminin güvenliğini arttırmak için dosya transferi aşamasında AES şifreleme kullanılmış, dosyalar parçalanarak aktarılmış ve IP katmanında temel düzeyde paket analizi yapılmıştır.

**2. PROJE AMACI VE KAPSAMI**

Projenin temel amacı, iki cihaz arasında (localhost üzerinden) çalışan, şifreli ve parçalanmış dosya transferi gerçekleştirebilen bir Python uygulaması geliştirmektir. Bu kapsamda geliştirilen modüller:

* AES şifreleme modülü
* Socket tabanlı dosya transfer sistemi
* Dosya parçalama ve birleştirme mekanizması
* IP paketi analiz modülü (Scapy kullanılarak)

**3. KULLANILAN YÖNTEMLER VE TEKNİK DETAYLAR**

**3.1. Socket ile Dosya Transferi**

Bu bölümde, sender.py ve receiver.py dosyaları kullanılarak temel bir TCP bağlantısı üzerinden dosya transferi gerçekleştirilmiştir.  
Aşağıda sender tarafında dosya gönderme, receiver tarafında dosya alma ve kaydetme kod parçaları gösterilmektedir  
Başarılı bir transfer sonrası “DOSYA GONDERILDI.” ve “DOSYA KAYDEDILDI.” mesajları alınmıştır.metin, yazı tipi, ekran görüntüsü, grafik içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**3.2. AES ile Şifreleme**

Dosya, gönderilmeden önce AES algoritmasının EAX modu kullanılarak şifrelenmiştir.  
Aşağıda şifreleme (**encrypt\_and\_digest**) ve çözümleme (**decrypt\_and\_verify**) işlemlerinin kod parçaları ile terminalde oluşan SHA-256 hash değerleri gösterilmiştir.  
Bu sayede verinin gizliliği ve doğruluğu sağlanmıştır.metin, yazı tipi, ekran görüntüsü, saat içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, yazı tipi, ekran görüntüsü, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**3.3. Parçalama ve Birleştirme**

Büyük verilerin ağ üzerinden güvenli ve hatasız aktarımı için dosya verisi 1024 baytlık parçalara bölünerek gönderilmiştir.  
Alıcı taraf bu parçaları sırasıyla birleştirerek orijinal veriyi oluşturmuştur.  
Aşağıda gönderim ve alım işlemleri sırasında kullanılan döngüler gösterilmiştir.

metin, yazı tipi, ekran görüntüsü, grafik içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**3.4. SHA-256 ile Hash Kontrolü**

Gönderilen ve alınan verilerin bütünlüğünü doğrulamak için SHA-256 algoritması kullanılarak özet değerler (hash) hesaplanmıştır.  
Terminalde elde edilen gönderici ve alıcı hash değerleri karşılaştırılmış ve eşleştiği görülmüştür.  
Bu, aktarım sırasında veri kaybı veya bozulması olmadığını kanıtlamaktadır.

metin, yazı tipi, ekran görüntüsü, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, yazı tipi, ekran görüntüsü, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.



metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.



metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.



**3.5. Scapy ile IP Paketi İncelemesi**

Scapy kütüphanesi kullanılarak oluşturulan örnek bir IP paketi üzerinden başlık bilgileri (version, ttl, flags gibi) analiz edilmiştir.  
Aşağıda IP paketinin oluşturulması ve packet.show() çıktısının terminal görüntüsü sunulmuştur.  
Böylece ağ katmanı seviyesinde veri paketlerinin yapısı gözlemlenmiştir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**4. GERÇEKLEŞTİRİLEN ADIMLAR VE GELİŞİM**

* Socket tabanlı dosya transferi başarıyla gerçekleştirilmiştir.
* AES şifreleme ve çözümleme işlevleri çalışır duruma getirilmiştir.
* Dosya parçalama ve birleştirme işlemleri sorunsuz tamamlanmıştır.
* SHA-256 hash kontrolü yapılmıştır.
* IP paketi Scapy üzerinden analiz edilmiş, başlık içeriği doğrulanmıştır.

Bu aşamada projede geliştirilen kod dosyaları da tamamlanmış ve proje klasörüne eklenmiştir. Eklenen kodlar:

* sender.py
* receiver.py
* scapy\_test.py

Bu dosyalar, projenin mevcut durumunu ve ilerlemesini belgelemek amacıyla rapora dahil edilmiştir.

**Projeyi Test Etme Yöntemi:**

* Terminal 1'de receiver.py dosyasını çalıştırarak bağlantı beklenir.
* Terminal 2'de sender.py çalıştırılarak dosya gönderimi başlatılır.
* Gönderilen dosyanın hash değeri ile alınan dosyanın hash değeri karşılaştırılır.
* Scapy testi için scapy\_test.py çalıştırılarak IP başlığı analizi yapılır.

**5. PROJE KALANINDA YAPILACAK ADIMLAR**

* Ping komutu ile ağ performansı testi yapılacak.
* Wireshark kullanılarak ağ trafiğindeki şifreli paketler gözlemlenecek.
* Kodların temizlenmesi ve yorumların eklenmesi sağlanacak.
* Final raporu hazırlanacak ve proje tamamlanacaktır.

**6. GITHUB PROJE DEPOSU**

Projeye ait kodlara aşağıdaki GitHub adresinden ulaşılabilirsiniz:

* <https://github.com/Yusuf-Guney/Bilgisayar-Aglari-Proje>

**7. KAYNAKLAR**

* [**https://docs.python.org/3/library/socket.html**](https://docs.python.org/3/library/socket.html)
* [**https://pycryptodome.readthedocs.io/en/latest/src/cipher/aes.html**](https://pycryptodome.readthedocs.io/en/latest/src/cipher/aes.html)
* [**https://scapy.readthedocs.io/en/latest/introduction.html**](https://scapy.readthedocs.io/en/latest/introduction.html)
* [**https://www.geeksforgeeks.org/sha-in-python/**](https://www.geeksforgeeks.org/sha-in-python/)