KRİPTOLOJİ DERSİ PROJE RAPORU

Okul Numarası: 436592

Ad - Soyad: Rahim Kaan Kacar

GitHub Repo: GitHub - rkaankacar/server-client-mesajla-ma

1. Sistem Başlatma ve Bağlantı Kurulması

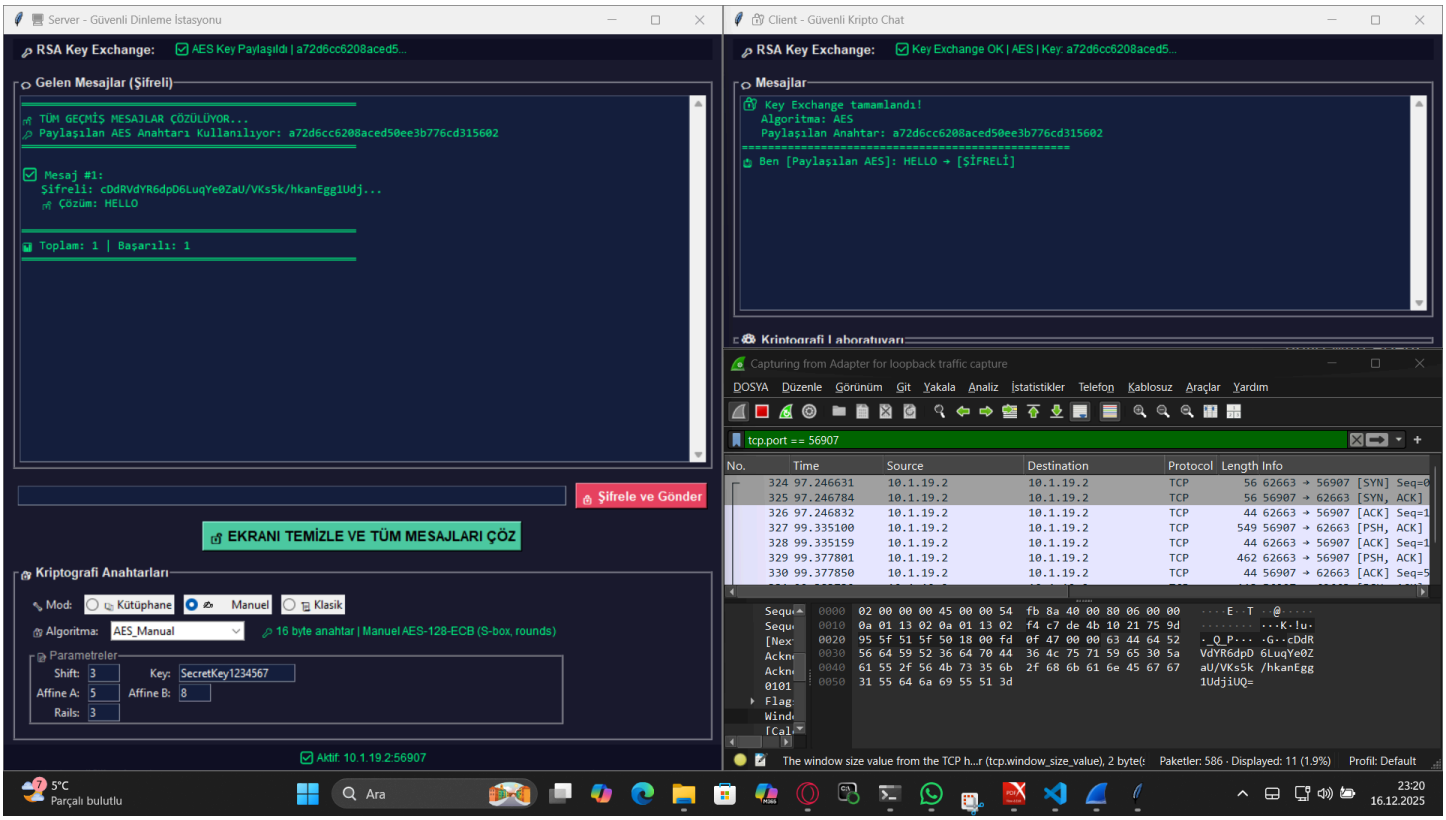
- **Sunucu (Server)**: main_server.py çalıştırıldığında, ServerUI sınıfı başlatılır. Sunucu, belirli bir IP adresi ve portta (otomatik olarak atanır) bağlantıları dinlemeye başlar. Tkinter tabanlı bir arayüz açılır ve sunucu hazır hale gelir.
- **İstemci (Client)**: main_client.py çalıştırıldığında, ClientUI sınıfı başlatılır. İstemci, sunucunun IP adresi ve portunu girerek bağlantı kurar. Bağlantı kurulduktan sonra, anahtar değişimi süreci başlar.

2. RSA Anahtar Değişimi (Key Exchange)

- Bu adım, güvenli bir simetrik anahtar paylaşımı için kullanılır. RSA asimetrik şifreleme algoritması ile gerçekleştirilir:
  - **Sunucu Tarafı**: Her yeni istemci bağlantısında, sunucu kendi RSA genel anahtarını (public key) istemciye gönderir. Bu, KeyExchangeServer sınıfı tarafından yönetilir.
  - **İstemci Tarafı**: İstemci, sunucudan gelen genel anahtarı alır ve rastgele bir simetrik anahtar (AES veya DES için) üretir. Bu anahtarı, sunucunun genel anahtarı ile şifreler ve geri gönderir. KeyExchangeClient sınıfı bu süreci yönetir.
  - **Tamamlanma**: Sunucu, kendi özel anahtarı (private key) ile şifreli anahtarı çözer. Artık hem sunucu hem istemci aynı simetrik anahtara sahip olur. Bu süreç "handshake" olarak adlandırılır ve tamamlandığında arayüzde "Key Exchange Tamamlandı" mesajı görünür.

3. Mesaj Gönderimi ve Şifreleme

- **Şifreleme Modları**: Sistem üç mod destekler:
  - **Kütüphane Modu**: AES_Library veya DES_Library gibi hazır kütüphane fonksiyonlarını kullanır.
  - **Manuel Mod**: AES_Manual veya DES gibi algoritmaların manuel implementasyonlarını kullanır.
  - **Klasik Mod**: Caesar, Vigenere, Rail Fence gibi geleneksel şifreleme yöntemlerini destekler (EncryptionFactory sınıfı üzerinden).



- **Mesaj Gönderimi**:
  - İstemci veya sunucu, mesajı girer.
  - Seçilen algoritma ve parametreler (örneğin, anahtar, shift değeri, rails sayısı) ile mesaj şifrelenir.
  - Şifreli mesaj, ağ üzerinden gönderilir ve karşı tarafta şifreli olarak görüntülenir.
  - Eğer anahtar değişimi tamamlanmışsa, paylaşılan simetrik anahtar kullanılır; aksi takdirde manuel ayarlar kullanılır.

4. Mesaj Alımı ve Çözme

- **Alım**: Gelen şifreli mesajlar, arayüzde "Şifreli" olarak gösterilir ve bir listede saklanır.
- **Çözme**: "EKRANI TEMİZLE VE TÜM MESAJLARI ÇÖZ" butonuna basıldığında:
  - Tüm biriken şifreli mesajlar, seçilen algoritma ve anahtar ile çözülür.
  - Başarılı çözümler arayüzde görüntülenir; hatalı olanlar işaretlenir.
  - Paylaşılan anahtar varsa, o kullanılır; yoksa manuel ayarlar uygulanır.

5. Ek Özellikler ve Güvenlik

- **Çoklu İstemci Desteği**: Sunucu, birden fazla istemciyi aynı anda yönetebilir. Her istemci için ayrı anahtar değişimi yapılır.
- **Güvenlik**: RSA ile anahtar paylaşımı güvenli olduğundan, mesajlar simetrik şifreleme ile hızlı ve güvenli bir şekilde iletilir.
- **Arayüz**: Tkinter ile geliştirilmiş, karanlık tema ile modern bir görünüm sunar. Algoritma seçimi, parametre ayarları ve durum göstergeleri bulunur.

Genel Akış Özeti:

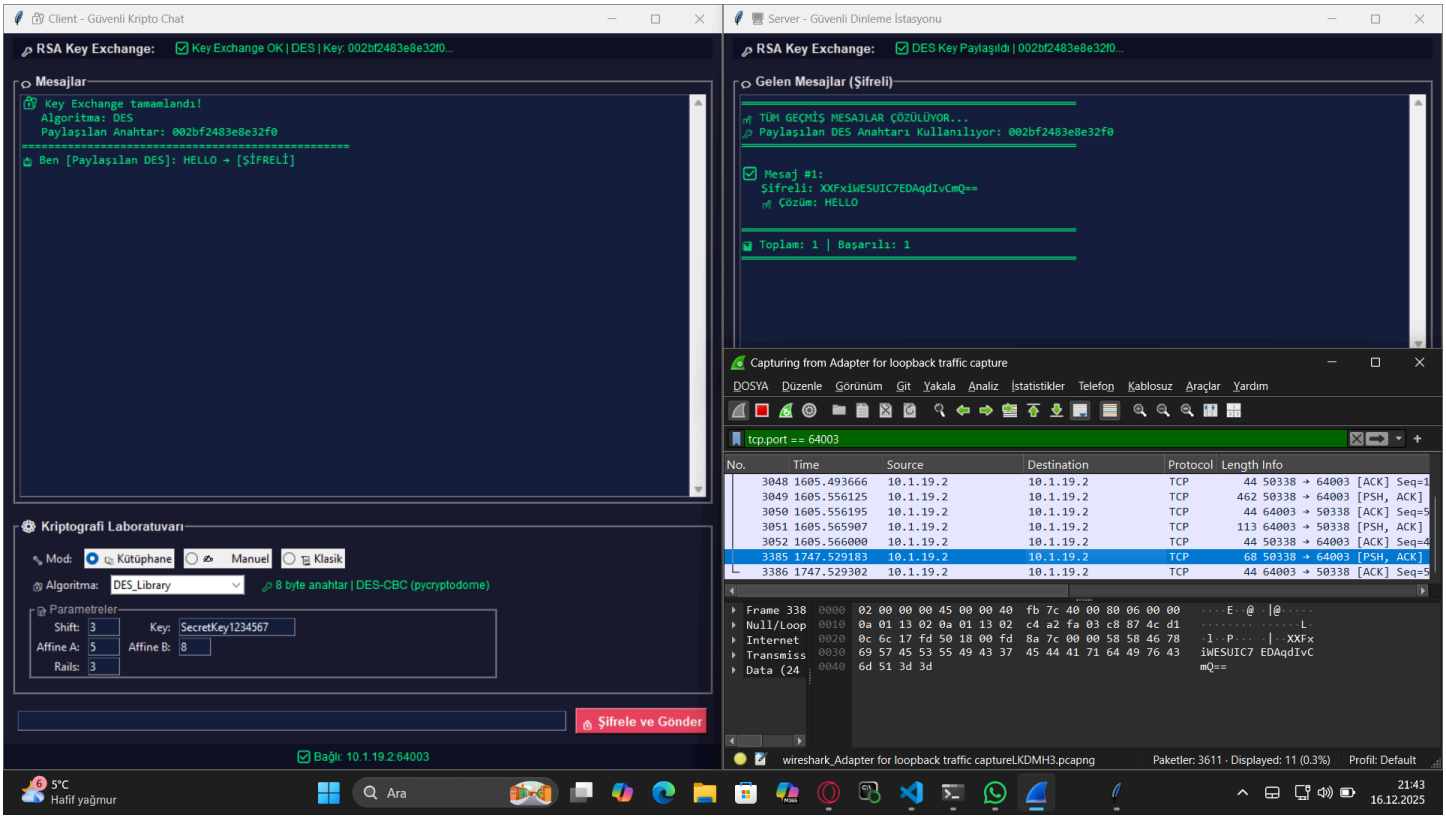
1. Sunucu başlatılır ve dinlemeye başlar.
2. İstemci bağlanır ve RSA anahtar değişimi gerçekleşir.

3. Mesajlar şifrelenerek gönderilir ve alınır.
4. İstenildiğinde tüm mesajlar çözülerek görüntülenir.

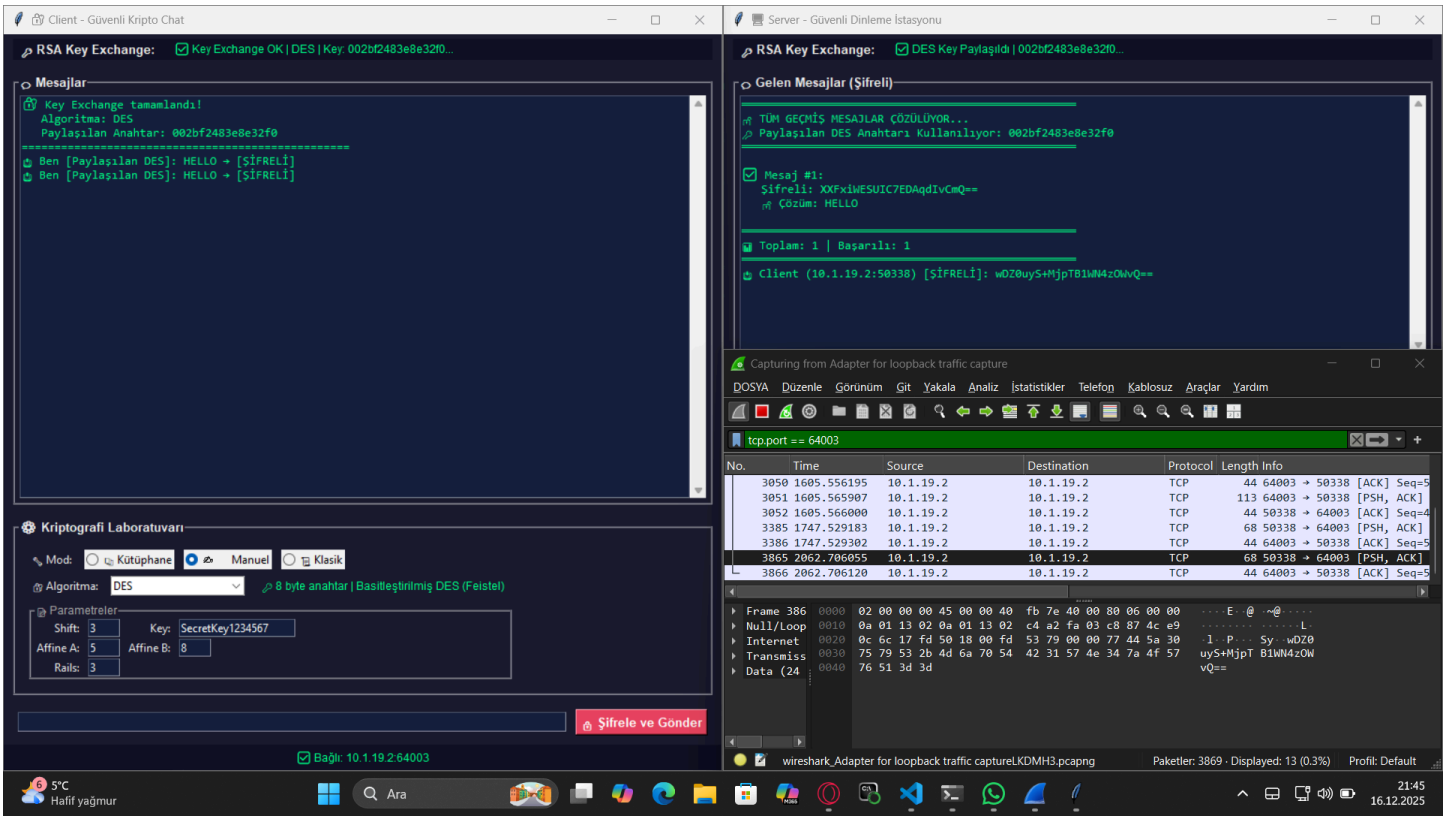
AES, DES ve RSA Karşılaştırması

Simetrik ve asimetrik olarak ikiye ayrılırlar. DES ve AES simetrik, RSA ise asimetriktir. DES 56 bittir, AES 128, 192 ve 256 bittir; RSA genelde 2048 bittir. DES brute force ile kırıldığı için yerine AES geliştirilmiştir. RSA, açık anahtar ve özel anahtar olmak üzere iki anahtar kullanır. Açık anahtar herkesle paylaşılabilirken, özel anahtar gizli tutulur. RSA yavaş olduğu için anahtar paylaşımında kullanılır. RSA ve AES birlikte kullanılır. Asıl veri AES ile, anahtar ise RSA ile şifrelenir. DES kırılabildiği için kullanılmamaktadır.

DES Kütüphane



DES Manuel

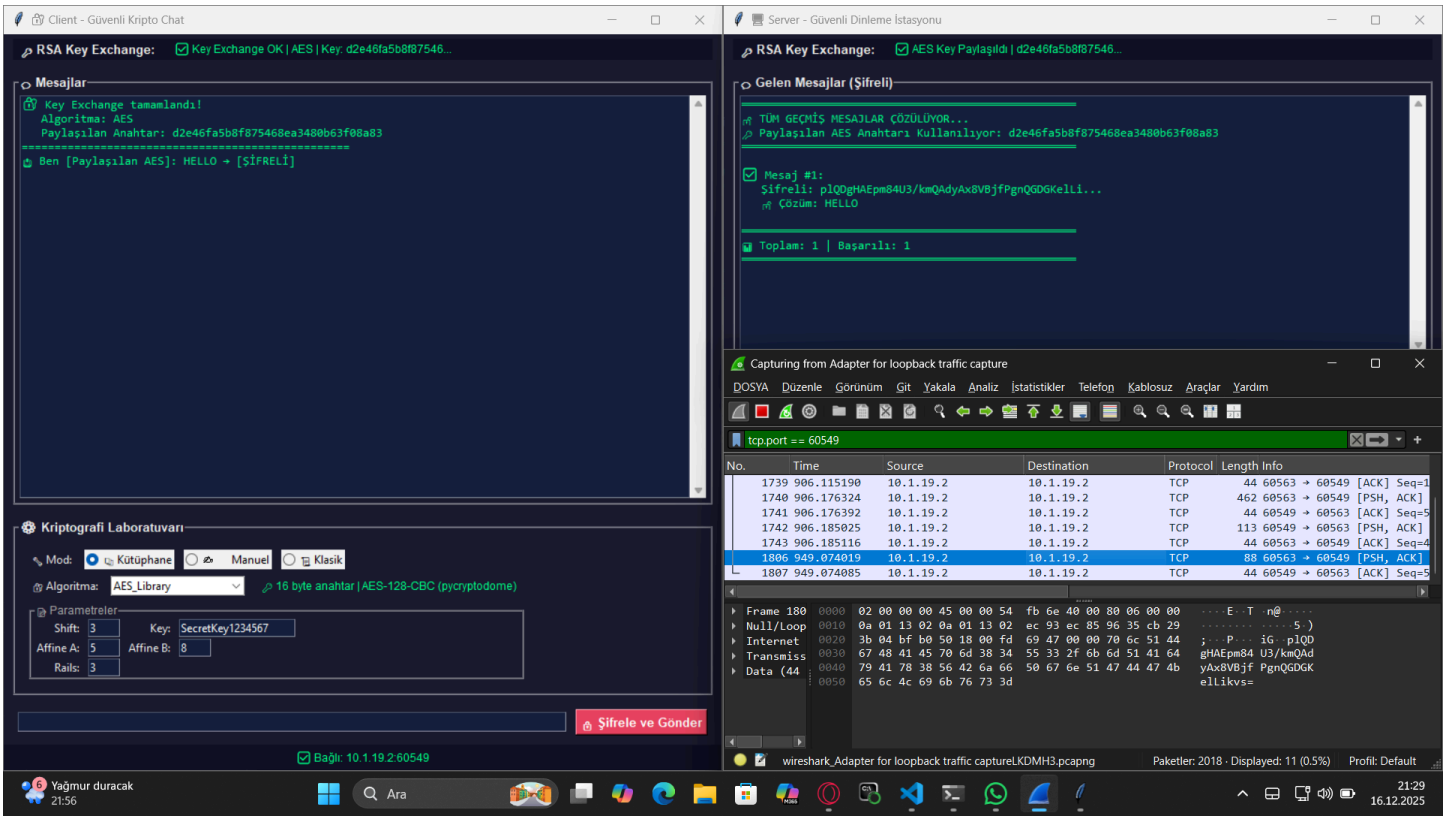


Wireshark ile DES algoritmasının ağ trafiği üzerindeki etkisi doğrulanmıştır. **HELLO** mesajı gönderilirken yakalanan paketlerin içeriği incelendiğinde, verinin şifreleme sayesinde açık metin olarak değil, seçilen DES algoritması tarafından üretilen şifreli metin şeklinde iletildiği ve veri gizliliğinin tam olarak sağlandığı gözlemlenmiştir. RSA ile güvenli anahtar değişimi ve DES ile veri şifreleme adımlarının ağ katmanında başarıyla uygulanmıştır. Manuel DES algoritmasının deşifrelemesi 10.222ms iken kütüphaneli DES deşifrelemesi 0.01ms'dir.

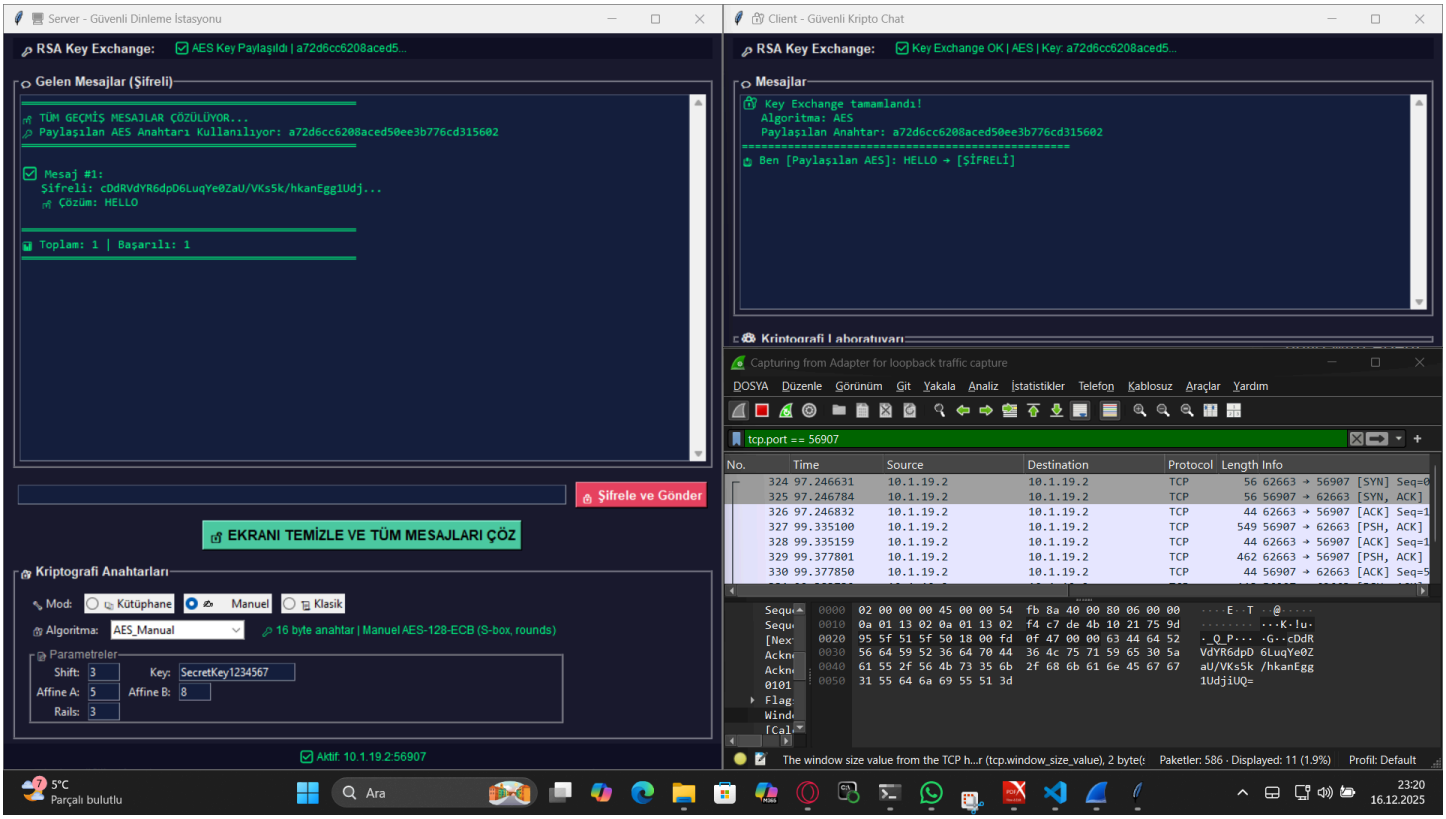
DES Manuel Öğrenci Yorumu

DES'in manuel olarak yazılması, Feistel yapısının çalışma mantığını anlamak açısından öğretici bir çalışma oldu. Algoritmanın sol ve sağ bloklara ayrılarak round fonksiyonu üzerinden ilerlemesi, simetrik şifrelemenin prensiplerini kavramamı sağladı. Round sayısının az ve yapının sade tutarak, algoritmayı basit ve anlaşılır halde yazdım. Manuel yazmak, DES'in temel mantığını öğrenmem için faydalı oldu; ancak gerçek projelerde güvenlik ve performans açısından daha modern şifreleme(AES) algoritmalarının tercih edilmesi gerektiğini gösterdi.

AES Kütüphane



AES Manuel



Manuel ve Kütüphaneli AES şifrelemelerin, Wireshark analizi sonucunda, istemci ve sunucu arasındaki iletişimin uçtan uca şifrelendiği doğrulanmıştır. Ağ üzerinden yakalanan paketler incelendiğinde, gönderilen **HELLO** mesajının açık metin olarak değil, seçilen AES algoritması tarafından üretilen şifreli metin şeklinde iletildiği ve veri gizliliğinin tam olarak sağlandığı gözlemlenmiştir. Ayrıca AES anahtarının ağ üzerinden açık bir şekilde gönderilmediği, RSA ile güvenli

bir şekilde paylaşılmıştır. Manuel AES algoritmasının deşifrelemesi 8.4ms iken kütüphaneli AES deşifrelemesi 0.05ms’dir.

AES Manuel Öğrenci Yorumu

AES-128 manuel yazılması, simetrik şifreleme algoritmalarının iç yapısını anlamak açısından oldukça öğretici bir çalışma oldu. AES’in temel adımları olan SubBytes, ShiftRows, MixColumns ve AddRoundKey işlemlerinin ayrı fonksiyonlar halinde yazılması, algoritmanın çalışma mantığını adım adım kavramamı sağladı. Ayrıca GF(2^8) aritmetiğinin fonksiyonlar aracılığıyla manuel olarak gerçekleştirilmesi, derslerde öğrenilen matematiksel altyapının pratikte nasıl kullanıldığını açıkça ortaya koymaktadır. Bu çalışma, AES’in mantığını öğrenmek için çok faydalı olup, gerçek projelerde uzunluk, karmaşıklık, güvenlik ve performans sebepleriyle hazır kriptografi kütüphanelerinin tercih edilmesi gerektiğini kavradım.