CLİENT-SERVER ŞİFRELEME UYGULAMASI

Numara: 436550

İsim: Emre Üçbudak

Github Repo: https://github.com/emreucbudak/KriptolojiOdev

1. Projenin Amacı ve Kapsamı

Bu çalışma, Kriptoloji dersinde öğrenilen teorik bilgileri pratik bir **Uçtan Uca Şifreli (E2EE)** haberleşme platformuna dönüştürmeyi amaçlamaktadır. Proje, sadece metin şifreleme işlevi görmemekte; aynı zamanda **Hibrit Şifreleme** mimarisini kullanarak ağ üzerinden güvenli anahtar değişimi ve çift yönlü mesajlaşmayı simüle etmektedir.

Projenin temel hedefleri:

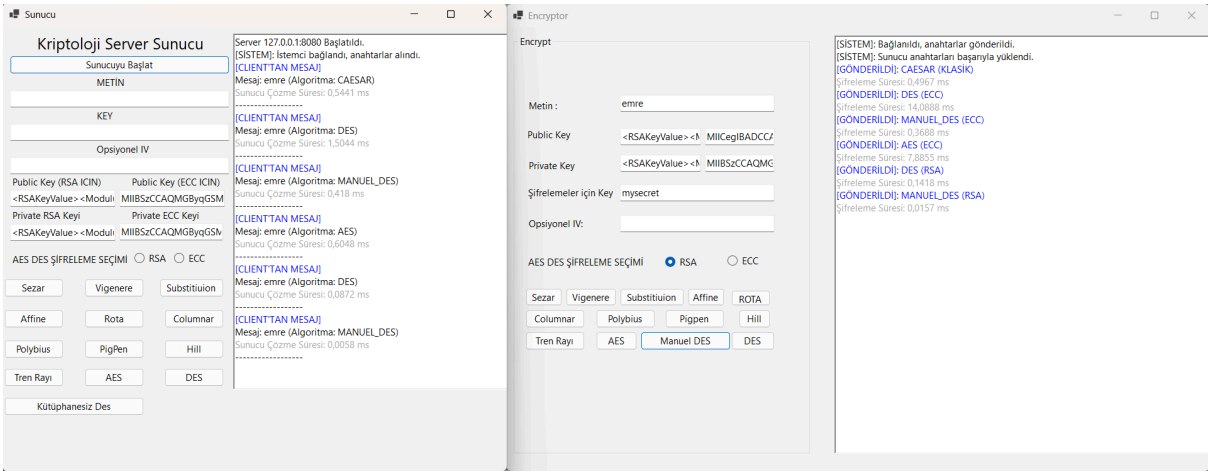
- **Asimetrik Şifreleme:** RSA ve modern **ECC (Elliptic Curve Cryptography)** kullanarak anahtar güvenliğini sağlamak.
- **Hibrit Mimari:** Simetrik algoritmaların (AES, DES) hızı ile asimetrik algoritmaların (RSA, ECC) güvenliğini birleştirmek.
- **Ağ Güvenliği:** TCP/IP soketleri üzerinden bir "Güvenli El Sıkışma" (Secure Handshake) protokolü geliştirmek.
- **Düşük Seviye Kodlama:** DES algoritmasını hiçbir kütüphane kullanmadan manuel olarak (bitwise işlemlerle) kodlayarak blok şifreleme mantığını kavramak.

---

2. Sistem Mimarisi ve Çift Yönlü İletişim

*Uygulama, yüksek performanslı ve gerçek zamanlı veri işleme yeteneğine sahip, çoklu iş parçacığı (multithreading) desteğiyle güçlendirilmiş bir **İstemci-Sunucu (Client-Server)** mimarisi üzerine inşa edilmiştir. Sistem, her iki tarafın da aktif olarak şifreli mesaj başlatabildiği çift yönlü bir iletişim protokolünü destekler.*

***Şekil 1:** Sunucu ve İstemci Uygulamalarının Genel Arayüz Görünümü (Bağlantı Kurulmuş Hal)*



2.1. Güvenli El Sıkışma (Handshake) ve Otomatik Anahtar Takası

*Bağlantı güvenliğini sağlamak amacıyla sistem, TCP oturumu açıldığı anda kullanıcı müdahalesine gerek duymayan otomatik bir **"Secure Handshake"** süreci başlatır:*

- ***Asimetrik Duyuru:** İstemci, sunucuya bağlandığı an kendi ürettiği **RSA-2048** ve **ECC (Secp256r1)** Public Key'lerini sunucuya iletir.*
- ***Bellek Kaydı ve Yanıt:** Sunucu, gelen bu anahtarları hafızasına alır ve kendi asimetrik Public Key'lerini istemciye göndererek yanıt verir.*
- ***Güvenli Kanalın Tesisi:** İstemci, sunucunun anahtarlarını aldığı anda kanal "Güvenli" statüsüne geçer ve hibrit şifreleme için gerekli olan asimetrik altyapı tamamlanmış olur.*

---

3. Kullanılan Algoritmalar ve Hibrit Güvenlik Mimarisi

Projede, modern siber güvenlik protokollerinin temelini oluşturan hibrit bir yapı kurgulanmıştır. Bu yapıda algoritmalar; veriyi şifreleyen "Simetrik Taşıyıcılar" ve anahtarı koruyan "Asimetrik Muhafızlar" olarak iki ana gruba ayrılır.

3.1. Simetrik Veri Taşıyıcılar (AES & DES)

Bu gruptaki algoritmalar, işlem hızları ve blok tabanlı yapıları sayesinde asıl mesaj içeriğinin şifrelenmesinden sorumludur.

- **AES (Advanced Encryption Standard): Projede 256-bit anahtar desteği ile yer alan AES, yüksek performanslı ve güvenli veri iletimi sağlar. Hibrit yapıda, mesajın kendisini şifrelemek için kullanılan ana algoritmadır.**
- **DES (Data Encryption Standard): 64-bit blok boyutu ile kurgulanan DES, hem kütüphane desteğiyle hem de eğitim amaçlı Manuel olarak implemente edilmiştir. Algoritmanın Feistel Ağı yapısı, bit seviyesindeki manipülasyonların gözlemlenmesi adına kritik bir rol oynar.**

3.2. Asimetrik Anahtar Muhafızları (ECC & RSA)

Asimetrik algoritmalar, simetrik anahtarların ağ üzerinden güvenli bir şekilde karşı tarafa ulaştırılmasını (Digital Envelope) sağlar.

- **ECC (Elliptic Curve Cryptography): Projenin en güncel bileşenidir. Secp256r1 eğrisi kullanılarak kurgulanmıştır. RSA'ya göre çok daha küçük anahtar boyutlarıyla (256-bit ECC ≈ 3072-bit RSA) aynı güvenlik seviyesini sunması, ağ trafiğinde paket yükünü minimize etmektedir.**

- **RSA (Rivest–Shamir–Adleman): 2048-bit anahtar uzunluğuyla kurgulanan RSA, sistemdeki alternatif asimetrik yöntemdir. Anahtar takası (Key Exchange) sırasında simetrik anahtarları kapsüllemek için kullanılır.**

3.3. Algoritmaların Karşılaştırmalı Analiz Tablosu

Özellik	AES	DES (Manuel/Lib)	RSA	ECC
Tür	Simetrik	Simetrik	Asimetrik	Asimetrik
Anahtar Boyutu	256 bit	64 bit	2048 bit	256 bit (Yüksek Verim)
Blok Boyutu	128 bit	64 bit	N/A (Zarf Tabl.)	N/A (Eğri Tabl.)
Hız	Çok Hızlı	Hızlı	Yavaş	Hızlı (RSA'dan Üstün)
Projedeki Rolü	Mesajı Şifreleme	Eğitim / Veri İletimi	Anahtar Dağıtımı	Güvenli Anahtar Zarfı

---

4. Manuel ve Kütüphane Tabanlı Şifreleme Analizi

Projenin akademik derinliğini pekiştirmek amacıyla, blok şifreleme sistemleri iki farklı yaklaşımla ele alınmıştır: Üst düzey kriptografik kütüphaneler (BCrypt, .NET Cryptography) ve bit seviyesinde manuel implementasyon.

4.1. Manuel DES (Kütüphanesiz) İmplementasyon Detayları

"Kütüphanesiz Şifreleme" gereksinimi doğrultusunda, `EncryptorService` sınıfı altında ManuelDesEncrypt metodu geliştirilmiştir. Hazır nesneler kullanılmadan, algoritmanın tüm mekanizması ham matematiksel ve mantıksal operatörlerle kodlanmıştır:

- **Blok ve Bit Yönetimi: Veri girişi tam 64-bitlik (8 byte) bloklara indirgenmiş; verinin her biti üzerinde `<<`, `>>`, `^` (XOR) ve `&` (AND) operatörleri kullanılarak doğrudan manipülasyon yapılmıştır.**
- **Feistel Ağı Yapısı: Algoritma, veriyi sol (L) ve sağ (R) olmak üzere ikiye bölen 16 döngülük bir Feistel Ağı üzerine kurulmuştur. Her round sonunda yer değiştirme ve karma işlemi manuel olarak tetiklenmiştir.**
- **Permütasyon ve S-Box Yönetimi: Standart DES tablosundaki Initial Permutation (IP), Substitution (S-Box) ve Permutation (P-Box) işlemleri dizi bazlı eşleşmelerle bit seviyesinde uygulanmıştır.**

4.2. Karşılaştırmalı Teknik Analiz

Analiz Kriteri	Kütüphane Tabanlı (AES/DES)	Manuel İmplementasyon (DES)
Performans	Optimize edilmiş C++ çekirdeği sayesinde milisaniyeler seviyesinde.	İşlemci üzerinde daha yüksek yük; akademik amaçlı kurgu.
Padding (Dolgulama)	PKCS7 gibi standart dolgulamalar otomatik yönetilir.	Blok boyutunu tamamlama işlemi manuel kodlanmıştır.
Güvenlik Katmanı	Yan kanal saldırılarına karşı önlemler (Timing attack protection) içerir.	Sadece temel matematiksel mantığı temsil eder.
Geliştirici Deneyimi	Kara kutu (Black Box) mantığıyla çalışır, iç yapı görünmez.	Beyaz Kutu (White Box); bitlerin nasıl yer değiştirdiği gözlemlenebilir.

---

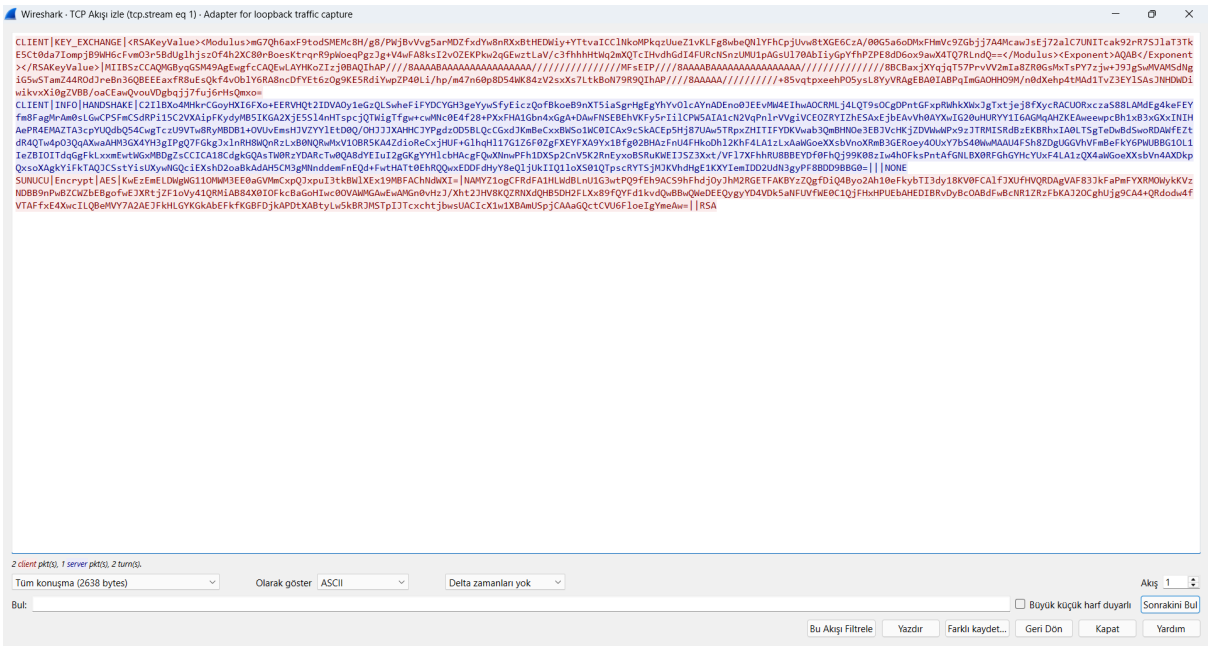
5. Wireshark ile Ağ Trafiği Analizi ve Paket İncelemesi

Sistemin TCP/IP protokolü üzerinden gerçekleştirdiği veri transferi, **Wireshark** ağ analiz aracı kullanılarak gerçek zamanlı olarak dinlenmiş ve elde edilen paketler üzerinde adli bilişim teknikleriyle inceleme yapılmıştır.

5.1. Hibrit Paket Yapısı ve Payload Gizliliği

Ağ üzerinden yakalanan paketler incelendiğinde, verinin tek bir blok halinde değil, kapsüllenmiş bir **"Hibrit Paket"** yapısında iletildiği görülmüştür. Paket içeriği;
`Operasyon|Algoritma|ŞifreliMetin|ŞifreliAnahtar|IV` bileşenlerinden oluşmaktadır.

- **Gizlilik Kanıtı:** Wireshark "Follow TCP Stream" özelliğinde, `TransportSecurity` katmanının etkisiyle veri tamamen anlamsız (ciphertext) karakter dizileri olarak görülmektedir. Bu durum, bir saldırganın ağdaki trafiği dinlemesi (Sniffing) durumunda dahi asıl mesajı veya bu mesajı çözen anahtarı (Key) asla elde edemeyeceğini teknik olarak kanıtlamaktadır.

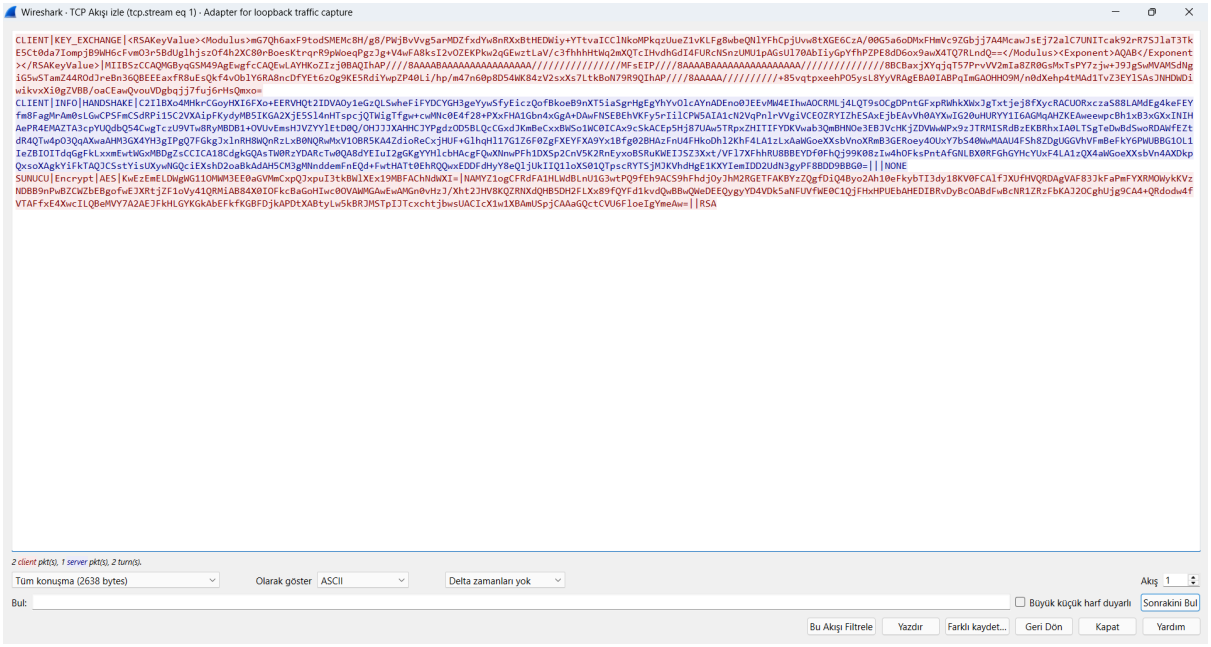


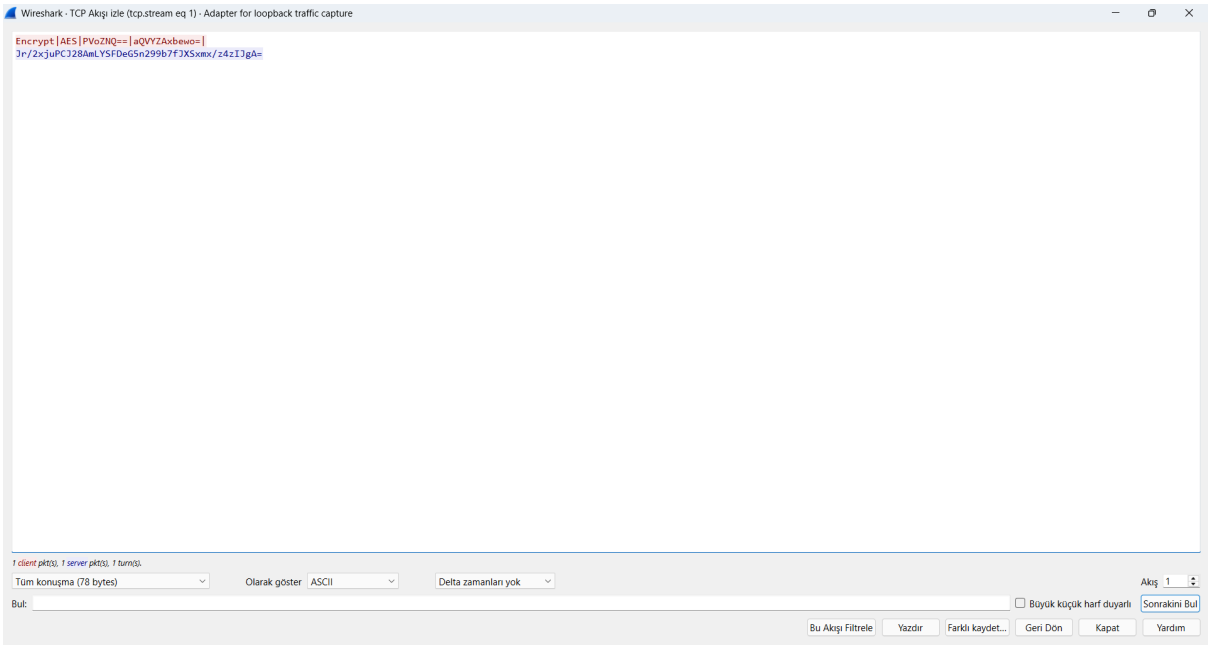
5.2. Paket Boyutlarının Karşılaştırması (AES/DES vs RSA)

Analiz sırasında en dikkat çekici fark, paket boyutlarında gözlemlenmiştir.

- **AES/DES: Blok boyutları küçük (128/64 bit) olduğu için, kısa metinlerde paket boyutu minimum seviyededir.**
- **RSA: 2048-bit anahtar kullanıldığında, şifreli çıktı (Ciphertext) doğrudan anahtarın modül boyutuna (256 byte) denk gelmektedir. Bu nedenle çok kısa bir metin ("Merhaba") bile şifrelense, RSA paket boyutu AES paket boyutuna göre oldukça büyüktür.**

Şekil 3: RSA Algoritmasının Paket Boyutuna Etkisi





Bu analiz, RSA'nın neden büyük verileri şifrelemek için değil, sadece küçük boyutlu AES anahtarlarını şifrelemek (Key Exchange) için kullanıldığını teknik olarak doğrulamaktadır.

---

6. Sonuç

Bu proje ile klasik kriptografiden modern hibrit yapılara uzanan geniş bir yelpaze, hem teorik hem de pratik düzeyde simüle edilmiştir. Sistemin statik bir şifreleme aracından, dinamik ve güvenli bir mesajlaşma platformuna dönüşme süreci şu teknik kazanımlarla sonuçlanmıştır:

- **Hibrit Şifreleme Başarısı: Veri gizliliğinin sağlanmasında simetrik algoritmaların (AES/DES) hızı ile asimetrik algoritmaların (ECC/RSA) güvenliği "Dijital Zarf" (Digital Envelope) mimarisi altında başarıyla birleştirilmiştir.**
- **Modern Asimetrik Yaklaşımlar: Geleneksel RSA-2048 metoduna alternatif olarak sunulan modern ECC (Secp256r1) implementasyonu ile daha düşük paket boyutlarında üst düzey güvenlik seviyelerine ulaşılmıştır.**
- **Otonom Güvenlik (Handshake): TCP katmanında kurgulanan otomatik el sıkışma protokolü sayesinde, anahtar takas sürecinin kullanıcı müdahalesi olmaksızın güvenli bir şekilde gerçekleştirilmesi sağlanmıştır.**
- **Düşük Seviye Algoritma Analizi: Manuel DES implementasyonu ile blok şifrelemenin yapı taşları (S-Box'lar, bitwise işlemler ve Feistel yapısı) yazılımsal olarak deneyimlenmiş, kütüphane bağımlılığı olmadan algoritma üretme yetkinliği kazanılmıştır.**
- **Ağ Katmanı Doğrulaması: Wireshark analizleri ile şifreli verinin ağ üzerindeki görünümü, paket boyutlarındaki değişimler ve asimetrik metotların (ECC vs RSA) ağ trafiği üzerindeki performans farkları somut verilerle ispatlanmıştır.**

Sonuç olarak bu çalışma; soket programlama, çoklu iş parçacığı yönetimi ve ileri düzey kriptografik metotların bir arada kullanıldığı, modern siber güvenlik standartlarını karşılayan başarılı bir güvenli haberleşme simülasyonu olmuştur.