

# KRİPTOGRAFI VE AĞ GÜVENLİĞİ TEKNİK RAPORU

## AES – DES – RSA Kullanarak İstemci–Sunucu Şifreleme Sistemi Analizi

### Öğrenci Bilgileri

- **Ad Soyad:** Hüseyin Bünyamin Gülme
- **Öğrenci No:** 439591
- **GitHub Deposu:** <https://github.com/huseyingulme/Kriptoloji>

### 1. Giriş ve Problemin Tanımı

Bilgi güvenliği, modern bilgisayar ağlarının en temel gereksinimlerinden biridir. Günümüzde bankacılık sistemleri, e-ticaret platformları, bulut tabanlı servisler ve IoT altyapıları büyük ölçüde istemci–sunucu mimarisi üzerinden çalışmaktadır. Bu mimaride istemci ile sunucu arasında sürekli olarak veri alışverişi gerçekleşmekte ve bu veriler çoğu zaman açık ağlar (internet gibi) üzerinden iletilmektedir.

Açık ağ ortamlarında gerçekleştirilen veri iletimi;

- **Pasif saldırılar** (dinleme, trafik analizi)
- **Aktif saldırılar** (veri değiştirme, paket enjeksiyonu)
- **Replay saldırıları**
- **Man-in-the-Middle (MITM)** saldırıları

gibi çok sayıda tehdit barındırmaktadır. Bu tehditlere karşı en temel savunma mekanizması kriptografidir.

Bu projede ele alınan temel problem şudur:

*Bir istemci ile sunucu arasında, ağ üzerinde dinlenmeye açık bir ortamda, veriler nasıl güvenli biçimde iletilebilir?*

Bu soruya cevap ararken, yalnızca “hangi algoritma daha güçlüdür?” yaklaşımı benimsenmemiş; aynı zamanda **hangi algoritma hangi amaç için uygundur, performans–güvenlik dengesi nasıl sağlanır ve ağ üzerindeki pratik etkiler nelerdir** gibi sorular da araştırılmıştır.

#### 1.1. Çalışmanın Amacı ve Kapsamı

Bu çalışmanın temel amaçları aşağıdaki şekilde özetlenebilir:

- Simetrik ve asimetrik kriptografik algoritmaların çalışma prensiplerini uygulamalı olarak incelemek
- AES, DES ve RSA algoritmalarını gerçek bir istemci–sunucu haberleşmesi üzerinde karşılaştırmak
- Hibrit şifreleme yaklaşımının neden modern sistemlerde zorunlu hale geldiğini göstermek
- Şifrelenmiş ağ trafiğinin Wireshark gibi araçlarla nasıl görüldüğünü analiz etmek
- Manuel kriptografik implementasyonlar aracılığıyla algoritmaların iç yapısını derinlemesine kavramak

Bu kapsamda proje, hem **kriptografi** hem de **ağ güvenliği** disiplinlerini birlikte ele alan bütüncül bir çalışma olarak tasarlanmıştır.

### 2. Sistem Mimarisi ve Tasarım Yaklaşımı

#### 2.1. İstemci–Sunucu Modelinin Seçilme Gerekçesi

İstemci–sunucu mimarisi, gerçek dünyada kullanılan güvenli iletişim sistemlerinin büyük çoğunluğunun temelini oluşturmaktadır. HTTPS, FTPS, SSH gibi protokoller bu mimari üzerine kuruludur. Bu nedenle projede seçilen mimari, teorik bilgilerin pratik karşılığını görmek açısından özellikle tercih edilmiştir.

TCP protokolü, aşağıdaki nedenlerle kullanılmıştır:

- Paket sıralaması ve yeniden iletim mekanizmaları
- Güvenilir bağlantı yapısı

- Uygulama katmanında güvenliğin test edilmesine olanak tanınması

Bu sayede kriptografik güvenlik, alt katman güvenliğinden bağımsız olarak ele alınabilmektedir.

## 2.2. Katmanlı Mimari Yaklaşım

Sistem, aşağıdaki mantıksal katmanlara ayrılmıştır:

1. **Kullanıcı Arayüzü Katmanı (GUI)**
2. **Paketleme ve İletişim Katmanı**
3. **Kriptografik İşlem Katmanı**
4. **Kontrol ve Yönlendirme Katmanı (ProcessingManager)**

Bu katmanlı yapı sayesinde:

- Kodun bakımı kolaylaşmış
- Güvenlik açıklarının izole edilmesi mümkün hale gelmiş
- Her katmanın sorumluluğu net biçimde ayrılmıştır

Bu yaklaşım, yazılım güvenliğinde önerilen **separation of concerns** ilkesine uygundur.

## 3. Veri Paketleme Stratejisi ve Güvenlik Etkileri

### 3.1. Metadata Kullanımının Önemi

Sadece verinin şifrelenmesi, güvenli iletişim için yeterli değildir. Alıcı tarafın:

- Hangi algoritmanın kullanıldığını
- Hangi anahtar ile çözümleme yapılacağını
- Paketin geçerli olup olmadığını

bilmesi gerekir. Bu nedenle DataPacket yapısı, **metadata destekli** olarak tasarlanmıştır.

Timestamp alanı özellikle dikkat çekicidir. Bu alan sayesinde:

- Aynı paketin tekrar gönderilmesi tespit edilebilmekte
- Replay saldırıları için temel bir savunma sağlanmaktadır

Gerçek dünyada TLS protokollerinde kullanılan **nonce** ve **sequence number** mekanizmaları ile benzer bir mantık izlenmiştir.

### 3.2. JSON Paketleme Modunun Akademik Katkısı

JSON tabanlı paketleme modu, doğrudan güvenliği artırmasa da **analiz edilebilirliği** ciddi ölçüde artırmıştır. Bu sayede:

- Şifreli veri ile kontrol bilgileri net biçimde ayrılmış
- Ağ analiz araçlarında paket içeriği daha anlaşılır hale gelmiştir
- Öğrenci açısından kriptografik süreçlerin gözlemlenmesi kolaylaşmıştır

Bu özellik, projeyi salt bir kodlama çalışması olmaktan çıkarıp, deneysel bir güvenlik analizine dönüştürmüştür.

## 4. Kriptografik Algoritmaların Derinlemesine İncelenmesi

### 4.1. AES'in Modern Kriptografideki Rolü

AES, günümüzde güvenli veri şifrelemenin fiili standardı haline gelmiştir. Bunun temel nedenleri:

- Güçlü matematiksel altyapı
- Donanım hızlandırma desteği (AES-NI)
- Uzun anahtar seçenekleri (128/192/256 bit)

Manuel implementasyon sürecinde özellikle şu noktalar dikkat çekmiştir:

- AES'in byte tabanlı çalışması
- Lineer ve lineer olmayan dönüşümlerin dengesi
- Difüzyon ve konfüzyon prensiplerinin başarılı uygulanması

Bu durum, AES'in neden DES'in yerini aldığını açıkça göstermektedir.

## 4.2. DES'in Eğitsel Değeri ve Güvenlik Açıkları

DES algoritması, kriptografi tarihinde önemli bir yere sahiptir ancak günümüz standartlarına göre yetersizdir. Projede yapılan denemeler, kısa anahtar uzunluğunun saldırgan açısından nasıl bir avantaj sağladığını somut biçimde ortaya koymuştur.

DES'in projeye dahil edilmesi sayesinde:

- Feistel ağ yapısı detaylı biçimde incelenmiş
- Simetrik şifreleme algoritmalarının evrimi anlaşılmış
- "Güvenli görünen" bir algoritmanın zamanla nasıl zayıflayabileceği görülmüştür

Bu durum, kriptografide **zaman faktörünün** ne kadar kritik olduğunu göstermektedir.

## 4.3. RSA ve Asimetrik Kriptografinin Sınırları

RSA, anahtar dağıtımı ve kimlik doğrulama için vazgeçilmezdir; ancak veri şifreleme için uygun değildir. Projede yapılan ağ analizleri, RSA'nın büyük anahtar boyutlarının ağ üzerinde ciddi bir yük oluşturduğunu açıkça ortaya koymuştur.

Bu gözlem, teorik olarak bilinen bir gerçeğin pratikte doğrulanmasını sağlamıştır:

*RSA güvenlidir, ancak pahalıdır.*

## 5. Hibrit Şifreleme Modelinin Stratejik Önemi

Hibrit şifreleme yaklaşımı, bu projenin en kritik kazanımlarından biridir. Bu yapı sayesinde:

- Asimetrik kriptografinin güvenli anahtar paylaşımı avantajı
- Simetrik kriptografinin yüksek performansı

bir araya getirilmiştir.

Bu yaklaşım, modern TLS protokollerinde kullanılan:

- RSA + AES
- ECDHE + AES

kombinasyonlarının mantığını birebir yansıtmaktadır.

Kriptoloji Projesi - Şifreleme/Çözme Sistemi

DosyaYardım

### Kriptoloji Projesi

Server Bağlantı Ayarları

Server IP: localhostPort: 12345BağlanBağlantıyı Kes

Metin İşlemeDosya İşlemeKayıtlı Dosyalar

Dosya: C:/Users/husey/Pictures/Screenshots/Ekran görüntüsü 2025-11-18 202415.pngDosya Seç

Dosya: Ekran görüntüsü 2025-11-18 202415.pngBoyut: 40893 bytesTip: imageDestekleniyor: Evet

Ayarlar

Algoritma: aesAlgoritma BilgisiÖrnek Anahtar? Deşifreleme Yardımı

Anahtar: secretkey12345678

☒ Şifrele☐ Çöz

Dosyayı İşle

İşlem Sonucu:

☒ Şifreleme tamamlandı!

Orjinal Dosya: Ekran görüntüsü 2025-11-18 202415.png

Şifreli Dosya: Ekran görüntüsü 2025-11-18 202415.png.enc

Otomatik Kayıt: EncryptedFiles\encrypted\Ekran görüntüsü 2025-11-18 202415.png.enc

Algoritma: aes



## 6. Wireshark Analizi ve Ağ Güvenliği Açısından Değerlendirme

Wireshark üzerinden yapılan incelemeler, kriptografinin ağ seviyesindeki etkilerini somutlaştırmıştır:

- Düz metin verinin tamamen gizlendiği
- Saldırganın yalnızca anlamsız byte dizileri görebildiği
- Paket boyutlarının kullanılan algoritmaya göre ciddi biçimde değiştiği

Bu analiz, “şifreleme yapılırca her şey güvenlidir” algısının yanlış olduğunu da göstermiştir. Yanlış algoritma seçimi, ağ performansını doğrudan olumsuz etkileyebilmektedir.

Capturing from Adapter for loopback traffic capture

OSYA Düzenle Görünüm Git Yakala Analiz İstatistikler Telefon Kablosuz Araçlar Yardım

tcp.port == 12345

No.	Time	Source	Destination	Protocol	Length	Info
433	45.397093	127.0.0.1	127.0.0.1	TCP	56	62719 → 12345 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
434	45.397168	127.0.0.1	127.0.0.1	TCP	56	12345 → 62719 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
435	45.397191	127.0.0.1	127.0.0.1	TCP	44	62719 → 12345 [ACK] Seq=1 Ack=1 Win=65280 Len=0
436	45.399516	127.0.0.1	127.0.0.1	TCP	154	62719 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=110
437	45.399537	127.0.0.1	127.0.0.1	TCP	44	12345 → 62719 [ACK] Seq=1 Ack=111 Win=65280 Len=0
438	45.401549	127.0.0.1	127.0.0.1	TCP	625	12345 → 62719 [PSH, ACK] Seq=1 Ack=111 Win=65280 Len=581
439	45.401599	127.0.0.1	127.0.0.1	TCP	44	62719 → 12345 [ACK] Seq=111 Ack=582 Win=64768 Len=0
476	55.194169	127.0.0.1	127.0.0.1	TCP	204	62719 → 12345 [PSH, ACK] Seq=111 Ack=582 Win=64768 Len=160
477	55.194214	127.0.0.1	127.0.0.1	TCP	44	12345 → 62719 [ACK] Seq=582 Ack=271 Win=65024 Len=0
478	55.217051	127.0.0.1	127.0.0.1	TCP	163	12345 → 62719 [PSH, ACK] Seq=582 Ack=271 Win=65024 Len=119
479	55.217103	127.0.0.1	127.0.0.1	TCP	44	62719 → 12345 [ACK] Seq=271 Ack=701 Win=64768 Len=0
559	61.816511	127.0.0.1	127.0.0.1	TCP	204	62719 → 12345 [PSH, ACK] Seq=271 Ack=701 Win=64768 Len=160
560	61.816553	127.0.0.1	127.0.0.1	TCP	44	12345 → 62719 [ACK] Seq=701 Ack=431 Win=65024 Len=0
561	61.832031	127.0.0.1	127.0.0.1	TCP	163	12345 → 62719 [PSH, ACK] Seq=701 Ack=431 Win=65024 Len=119
562	61.832060	127.0.0.1	127.0.0.1	TCP	44	62719 → 12345 [ACK] Seq=431 Ack=820 Win=64512 Len=0
790	91.836305	127.0.0.1	127.0.0.1	TCP	44	12345 → 62719 [FIN, ACK] Seq=820 Ack=431 Win=65024 Len=0
791	91.836328	127.0.0.1	127.0.0.1	TCP	44	62719 → 12345 [ACK] Seq=431 Ack=821 Win=64512 Len=0

Frame 561: Packet, 163 bytes on wire (1304 bits), 163 bytes captured (1304 bits) on interface Null/Loopback

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 12345, Dst Port: 62719, Seq: 701, Ack: 431, Len: 119

Source Port: 12345

Destination Port: 62719

[Stream index: 48]

[Stream Packet Number: 14]

[Conversation completeness: Incomplete, DATA (15)]

[TCP Segment Len: 119]

Sequence Number: 701 (relative sequence number)

Sequence Number (raw): 2033311702

[Next Sequence Number: 820 (relative sequence number)]

Acknowledgment Number: 431 (relative ack number)

Acknowledgment number (raw): 3336142198

0101 .... = Header Length: 20 bytes (5)

Flags: 0x018 (PSH, ACK)

Window: 254

[Calculated window size: 65024]

[Window size scaling factor: 256]

Checksum: 0x1316 [correct]

[Checksum Status: Good]

0000 02 00 00 00 45 00 00 9f 40 a3 40 00 80 06 00 00 .....E...@...  
0010 7f 00 00 01 7f 00 00 01 30 39 f4 ff 79 31 df d6 .....09..y1..  
0020 c6 d9 7d 76 50 18 00 fe 13 16 00 00 7b 22 6f 70 ...}vP... ..{"op  
0030 65 72 61 74 69 6f 6e 22 3a 20 22 52 45 53 55 4c eration": "RESUL  
0040 54 22 2c 20 22 6d 65 73 73 61 67 65 22 3a 20 22 T", "mes sage": "  
0050 73 65 6c 61 6d 6c 61 72 22 2c 20 22 61 6c 67 6f selamlar ", "algo  
0060 72 69 74 68 6d 22 3a 20 22 63 61 65 73 61 72 22 rithm": "caesar"  
0070 2c 20 22 73 75 63 63 65 73 73 22 3a 20 74 72 75 , "succe ss": tru  
0080 65 2c 20 22 74 69 6d 65 73 74 61 6d 70 22 3a 20 e, "time stamp":  
0090 31 37 36 35 35 37 34 39 32 38 2e 30 38 35 36 37 17655749 28.08567  
00a0 31 32 7d 12}

## 7. Genel Değerlendirme ve Öğrenilen Dersler

Bu proje sürecinde elde edilen en önemli kazanımlar şunlardır:

- Kriptografi yalnızca matematik değil, aynı zamanda sistem tasarımıdır
- Güvenlik, tek bir algoritma ile değil, bütüncül bir mimari ile sağlanır
- Manuel implementasyonlar, algoritmaların gerçek doğasını anlamak için vazgeçilmezdir
- Ağ trafiği analiz edilmeden yapılan güvenlik iddiaları eksik kalır

## 8. Sonuç

Bu çalışma, güvenli istemci–sunucu haberleşmesinin yalnızca güçlü algoritmalarla değil; **doğru mimari, hibrit şifreleme, paketleme stratejileri** ve **ağ seviyesi analizlerle** mümkün olduğunu göstermiştir. Proje, teorik kriptografi bilgisini pratik ağ güvenliği uygulamalarıyla birleştiren kapsamlı ve öğretici bir çalışma olarak tamamlanmıştır.