

PROJE UYGULAMA RAPORU: AŞAMA 2 - KODLAMA VE TEST Tarih: 19.12.2025

2.1 Geliştirme Ortamı ve Dil Seçimi

MOD-64 algoritmasının kodlanması aşamasında **Python** programlama dili tercih edilmiştir. Python'un seçilme nedenleri şunlardır:

- **Bit Seviyesi İşlemler:** Python, büyük tamsayılar üzerinde bit kaydırma (`<<`, `>>`) ve mantıksal operatörleri (XOR, AND) doğrudan ve hızlı bir şekilde uygulayabilmektedir.
- **Okunabilirlik:** Feistel Ağrı mimarisinin karmaşık mantığını sade ve anlaşılır bir sözdizimi ile ifade etmeye olanak tanır.
- **Geliştirme Ortamı (IDE):** Kodlama işlemi **Visual Studio Code (VS Code)** ortamında gerçekleştirılmıştır.

Geliştirilen Temel Fonksiyonlar:

Algoritma, nesne tabanlı bir yapı içinde aşağıdaki 3 ana fonksiyon üzerine inşa edilmiştir:

1. **Anahtar_Uret(parola):** Kullanıcıdan alınan string tipindeki parolayı ASCII dönüşümü ile 64-bitlik bir tamsayıya çevirir. Ardından her tur (8 tur) için bit kaydırma yöntemiyle özgün alt anahtarlar (Round Keys) üretir.
2. **Sifrele(duz_metin, anahtar):** 64-bitlik düz metin bloğunu alır. 8 turluk Feistel döngüsüne sokar. Her turda sağ bloğu F-Fonksiyonu (XOR + S-Kutusu + Permütasyon) ile işleyip sol blokla XOR işlemine tabi tutar. Sonuçta şifreli sayısal veriyi (ciphertext) döndürür.
3. **Desifrele(sifreli_metin, anahtar):** Şifreli veriyi alır. Şifreleme işleminin aynısını uygular, ancak bu sefer Anahtar_Uret fonksiyonundan gelen anahtarları **ters sırada** kullanır. Bu sayede işlem matematiksel olarak geri alınır.

2.2 Test ve Doğrulama

Algoritmanın güvenilirliğini ve Feistel mimarisinin doğru çalıştığını kanıtlamak amacıyla iki kritik test senaryosu uygulanmıştır.

Test 1: Basit Doğrulama (Correctness)

- **Amaç:** Verilen düz metnin şifrelendikten sonra doğru anahtarla tekrar eski haline döndürülebildiğini kanıtlamak.
- **Girdi:** Metin: "KALE", Anahtar: "AN"
- **Beklenen Sonuç:** Deşifreleme çıktısının "KALE" olması.

- **Gerçekleşen Sonuç:** Algoritma metni başarıyla şifrelemiş ve şifreli veriyi doğru anahtarla çözerek "KALE" çıktısını vermiştir.
- **Durum:** ✅ BAŞARILI

Test 2: Anahtar Hassasiyeti (Avalanche Effect)

- **Amaç:** Anahtardaki çok küçük bir değişikliğin (1 bit veya 1 harf), şifre çözme işlemini tamamen başarısız kıldığını göstermek (Kriptografik Çığ Etkisi).
- **Senaryo:**
 1. Metin "AN" anahtarı ile şifrelendi.
 2. Şifreli veri, "AM" anahtarı ile (Sadece 1 harf farklı) çözülmeye çalışıldı.
- **Beklenen Sonuç:** Anlamsız veya bozuk bir metin çıkması.
- **Gerçekleşen Sonuç:** Sistem, yanlış anahtar verildiğinde orijinal metin yerine [Anlamsız Veri] veya bozuk karakterler üretmiştir.
- **Durum:** ✅ BAŞARILI

Sonuç: Gerçekleştirilen kodlama ve testler sonucunda, MOD-64 algoritmasının Feistel mimarisine uygun olarak çalıştığı, şifreleme-deşifreleme döngüsünü tamamladığı ve anahtar değişikliklerine karşı duyarlı olduğu doğrulanmıştır.

```

main.py
19     return sonuc
20
21 def _f_fonksiyonu(self, sag_blok, tur_anahtari):
22     """F-Fonksiyonu: XOR + S-Kutusu + Permütasyon"""
23     t = sag_blok ^ tur_anahtari
24     s_cikti = self._s_kutusu(t)
25     return self._dairesel_sola_kaydir(s_cikti, 5)
26
27 def Anahtar_Uret(self, parola):
28     """
29         İster 1: Paroladan 8 adet tur anahtarı üretir.
30     """
31     parola = parola.ljust(8)[:8] # 64-bit tamamla
32     ana_anahtar = int.from_bytes(parola.encode('utf-8'),
33                                   'big')
34
35     anahtarlar = []
36     temp_key = ana_anahtar
37     for i in range(8):
38         tur_anahtari = temp_key & 0xFFFFFFFF
39         anahtarlar.append(tur_anahtari)
40         # Anahtarı karıştır (Key Schedule)
41         temp_key = ((temp_key << 7) | (temp_key >> (64 - 7
42             ))) & ((1 << 64) - 1)

```

Output

```

--- TEST 1: BASIT DOĞRULAMA ---
Giriş Metni: KALE
Anahtar: AN
Şifreli Veri (Hex): 0x98a63a9e4f5b9dc2
Çözülen Metin: KALE
SONUÇ: ✅ Test 1 Başarılı (Metin geri döndürüldü)

--- TEST 2: ANAHTAR HASSASIYETİ (ÇİĞ ETKİSİ ---
Orijinal Şifreli Veri: 0x98a63a9e4f5b9dc2
Yanlış Anahtar: AM (Sadece 1 bit/harf farklı)
Yanlış Anahtarla Çözülen: [Anlamsız Veri]
SONUÇ: ✅ Test 2 Başarılı (Yanlış anahtar veriyi açamadı)

== Code Execution Successful ==

```