T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ ANABİLİM DALI
SİBER GÜVENLİK BİLİM DALI

SG 503 KRİPTOLOJİ DERSİ PROJE RAPORU

Hazırlayanlar
Y225012153 – Ayşenur ÖZKAN
Y225012150 – Sena ASLIBAY

Dersin Öğretim Görevlisi
Doç. Dr. Ünal ÇAVUŞOĞLU

2022- 2023 Bahar Dönemi

İÇİNDEKİLER

1. ALGORİTMADA KULLANILAN YÖNTEMLER

1.1. Feistel Mimarisi

Feistel mimarisi, blok şifrelerin yapımında kullanılan simetrik bir yapıdır. Bu yapıda, şifrelenecek veri iki eşit parçaya bölünür ve her parça bir raund fonksiyonu ile şifrelenir. Raund fonksiyonu, veri parçasını ve bir alt anahtarı alarak bir çıktı üretir. Bu çıktı diğer veri parçası ile XOR işlemine tabi tutulur. Bu işlem belirli sayıda tekrarlanır ve son çıktı şifreli veridir. Şifre çözme işlemi ise şifreleme işleminin tersi olarak yapılır. Feistel mimarisi, şifreleme ve şifre çözme işlemlerinin çok benzer olması, hatta bazı durumlarda aynı olması avantajına sahiptir.

1.2. Blok Şifreleme

Blok şifreleme, simetrik anahtar kullanarak belirli bir algoritmayla sabit uzunluktaki bit gruplarını (blokları) şifreleyen veya şifresini çözen bir kriptografi yöntemidir. Blok şifreleme, birçok kriptografik protokolün temelini oluşturur ve büyük verilerin şifrelenmesinde sıkça kullanılır.

1.3. Simetrik Şifreleme

Simetrik şifreleme, aynı gizli anahtarla hem veriyi şifreleyen hem de şifresini çözen bir şifreleme yöntemidir . Simetrik şifreleme, kriptografi teknikleri ve şifreleme algoritmaları arasında en eski ve en yaygın olanıdır . Gizli anahtar, sayısal veya sözel bir değer veya rastgele karakterlerden oluşabilir . Simetrik şifreleme algoritmaları aynı anahtarla hem şifreleme hem de şifre çözme yapar. Simetrik şifrelemenin avantajları arasında hızlı işlem, kolay uygulanabilirlik ve veri gizliliği bulunur . Simetrik şifrelemenin dezavantajları arasında ise anahtar saklama, anahtar dağıtma, kimlik doğrulama ve bütünlük kontrolü sorunları yer alır.

1.4. CBC ve ECB Nedir

Blok şifreleme modları, bir blok şifresinin düz metinleri nasıl şifrelediğini belirleyen yöntemlerdir. Bir blok şifresi, bir anahtar yardımıyla belirli bir boyuttaki düz metinleri şifreli metinlere çeviren bir algoritmadır.

ECB (Elektronik Kod Kitabı), en basit blok şifreleme modudur. Bu modda, düz metin blokları anahtarla doğrudan şifrelenir ve aynı boyutta şifreli metin blokları elde edilir . Bu modun sorunu, aynı düz metin bloğunun her seferinde aynı şifreli metin bloğuna dönüşmesidir. Bu da şifreli metinde tekrar eden desenler oluşmasına ve saldırganların bunlardan yararlanmasına neden olur.

CBC (Şifreleme Blok Zincirleme), ECB'nin bu sorununu çözmek için geliştirilmiş bir moddur. Bu modda, her düz metin bloğu önce kendisinden önce gelen şifreli metin bloğuyla XOR işlemine sokulur ve sonra anahtarla şifrelenir . Böylece, aynı düz metin bloğu farklı durumlarda farklı şifreli metin bloklarına dönüşebilir. İlk düz metin bloğu için ise rastgele seçilmiş bir başlatma vektörü (IV) kullanılır. CBC modu, ECB'ye kıyasla daha güvenli ve popüler bir moddur.

1.5. Anahtar Üretimi

Anahtar üretimi, kriptografik algoritmaların güvenliğinin temelini oluşturan bir süreçtir. Anahtar üretimi, bir anahtarın rastgele veya yarı-rastgele olarak oluşturulması ve dağıtılması işlemidir. Anahtarın boyutu, karmaşıklığı ve tahmin edilemezliği gibi özelliklere bağlı olarak farklı yöntemlerle yapılabilir. Anahtar üretimi, saldırganların anahtarı ele geçirmesini veya kırmasını zorlaştırmak için güvenli ve verimli olmalıdır. Simetrik şifrelemede, aynı anahtar hem şifreleme hem de şifre çözme için kullanılır ve taraflar arasında gizli tutulmalıdır.

2. ŞİFRELEME ALGORİTMASININ GELİŞTİRİLMESİ

2.1. Şifreleme İşleminde Kullanılacak Değişkenlerin Değerlerinin Belirlenmesi

ROUNDS, BLOCKSIZE, BLOCKSIZE_BITS, PATH_TO_FILES ve SECRET adında beş sabit tanımlanacak. Bu sabitlerin değerleri şöyle:

ROUNDS = 8, şifreleme ve deşifreleme işlemlerinde kullanılacak tur sayısını belirtir.

BLOCKSIZE = 8, şifreleme ve deşifreleme işlemlerinde kullanılacak blok boyutunu bayt cinsinden belirtir.

BLOCKSIZE_BITS = 64, şifreleme ve deşifreleme işlemlerinde kullanılacak blok boyutunu bit cinsinden belirtir.

PATH_TO_FILES = os.getcwd()+"/", dosyaların bulunduğu dizinin yolunu belirtir.

SECRET = "3f788083-77d3-4502-9d71-21319f1792b6", şifreleme anahtarını oluşturmak için kullanılacak gizli bir dizgedir.

2.2. Kullanıcının Algoritmayı Çalıştırması

Kodun bir sonraki kısmında main adında bir fonksiyon tanımlıyor. Bu fonksiyon, algoritmaya verilen argümanları ayrıştırarak şifreleme veya deşifreleme modunu, kriptografik modu, girdi dosyasını, anahtarı ve çıktı dosyasını belirler. Ardından, encryptMessage veya decryptCipher fonksiyonlarını çağırarak girdi dosyasındaki mesajı şifreler veya deşifreler ve ilgili çıktıyı üreterek ekrana yazar.

2.3. EncryptMessage Fonksiyonunun Oluşturulması

Kodun bu bölümünde encryptMessage adında bir fonksiyon tanımlıyor. Bu fonksiyon, verilen anahtar, mesaj ve mod ile mesajı şifrelemek için Feistel ağı adlı bir algoritma kullanılarak oluşturulmuştur. Feistel ağı, mesajı bloklara ayırır ve her bloğu ROUNDS sayısı kadar tekrarlanan bir işlemden geçirir. Bu işlemde, bloğun sol yarısı sağ yarısıyla değiştirilir, sağ yarısı ise sol yarısıyla XOR işlemine tabi tutulur. XOR işlemi, iki dizgenin aynı uzunlukta olması halinde her bir karakterin ikili değerini karşılaştırır ve farklıysa 1, aynıysa 0 döndürür. Örneğin, "1010" ile "1100" XOR işlemine tabi tutulursa "0110" sonucu çıkar. Sol yarısı, scramble adında bir başka fonksiyonla karıştırılır. Bu fonksiyon, sağ yarısı, tur sayısı ve anahtarın ikili değerlerini alır ve bunları çarpıp üs alarak bir sonuç üretir. Bu sonuç tekrar ikiliye çevrilir ve sol yarısıyla XOR işlemine tabi tutulur. Bu işlem sonunda, şifreli mesajın sol ve sağ yarısı birleştirilir ve çıktı olarak döndürülür.

2.4. DecryptCipher Fonksiyonunun Oluşturulması

Kodun bu kısmında decryptCipher adında bir fonksiyon tanımlıyor. Bu fonksiyon, verilen anahtar, şifreli mesaj ve mod ile şifreli mesajı deşifrelemek için Feistel ağının tersini uygular. Yani, şifreli mesajı bloklara ayırır ve her bloğu ROUNDS sayısı kadar tekrarlanan bir işlemden geçirir. Bu işlemde, bloğun sağ yarısı sol yarısıyla değiştirilir, sol yarısı ise sağ yarısıyla XOR işlemine tabi tutulur. Sağ yarısı, scramble fonksiyonuyla karıştırılır ve sol yarısıyla XOR işlemine tabi tutulur. Bu işlem sonunda, deşifreli mesajın sol ve sağ yarısı birleştirilir ve çıktı olarak döndürülür.

2.5. Key_256 Fonksiyonunun Oluşturulması

Key_256 adındaki bu fonksiyon, verilen anahtar ile SECRET dizgesini birleştirir ve SHA-256 algoritmasıyla şifreler. SHA-256 algoritması, herhangi bir uzunluktaki bir dizgeyi alarak 256 bitlik bir özet üretir. Bu özet, dizgenin benzersiz bir tanımlayıcısıdır ve şifreleme için kullanılır. Fonksiyon, özetin onaltılık değerini döndürür.

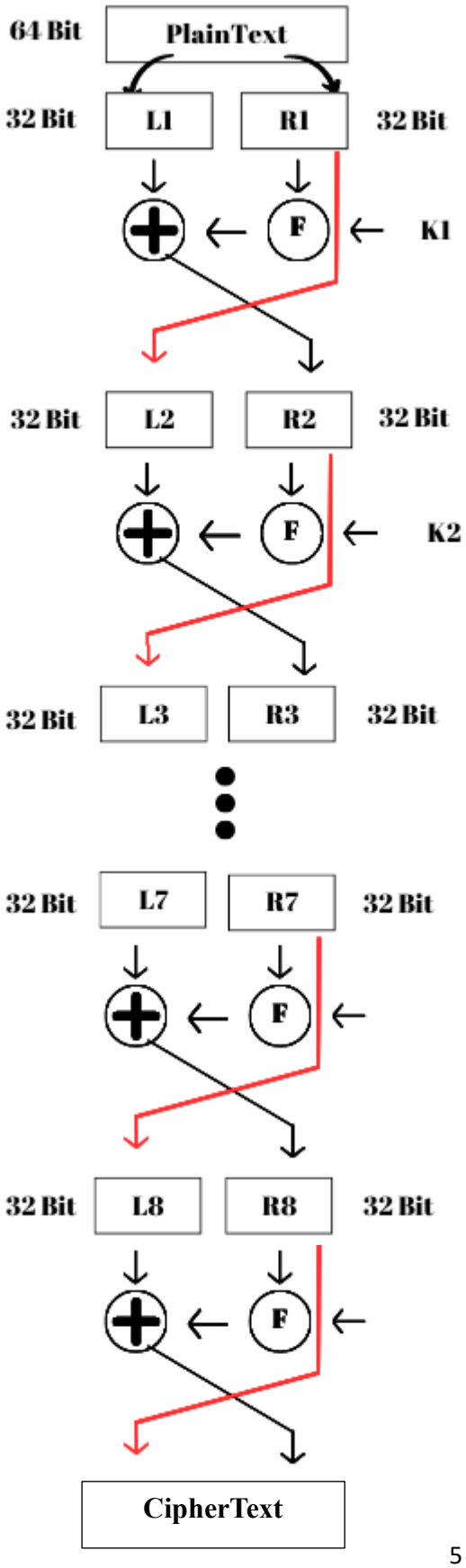
2.6. Subkeygen Fonksiyonunun Oluşturulması

Kodun bu bölümünde subkeygen adında bir fonksiyon tanımlıyor. Bu fonksiyon, verilen iki dizge ve bir sayıyı alarak yeni bir anahtar üretir. Aynı zamanda iki dizgeyi birleştirir ve SHA-256 algoritmasıyla şifreler. Ardından, özetin onaltılık değerini döndürür. Bu fonksiyon, CBC modunda şifreleme yaparken her turda farklı bir anahtar kullanmak için oluşturulmuştur.

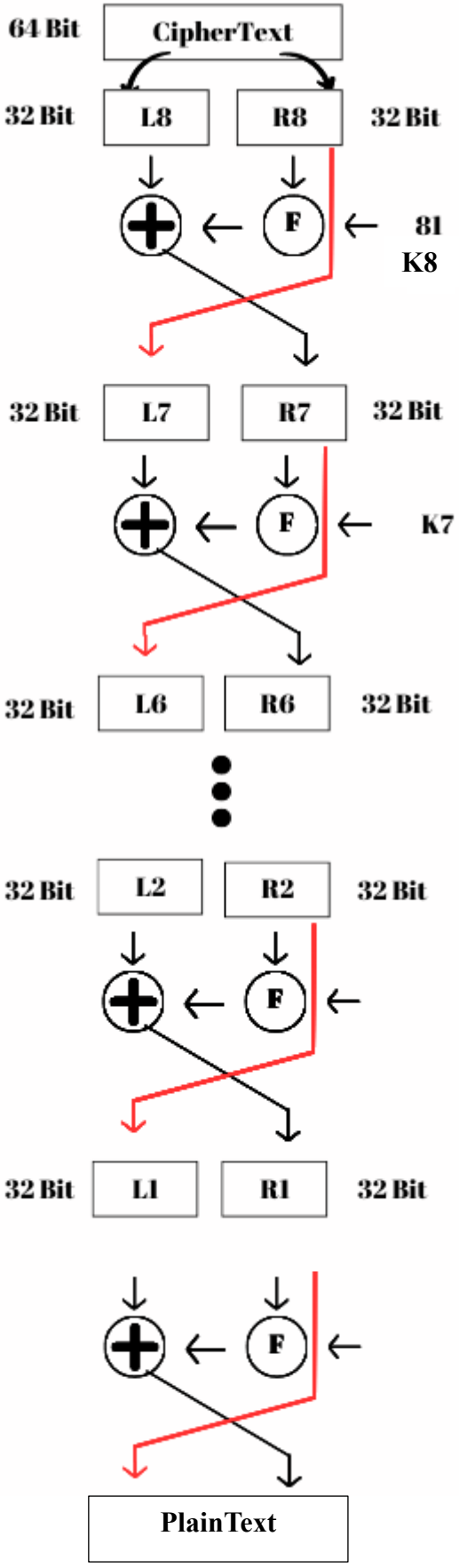
2.7. Scramble Fonksiyonunun Oluşturulması

Scramble adındaki bu fonksiyon, verilen bir dizge, bir sayı ve bir anahtarı alarak dizgeyi karıştırır. Bu fonksiyon, dizge ve anahtarı ikilik tabana çevirir ve bunların ikili değerlerini alır. Ardından, ikilik tabandaki değerleri çarpar ve sayının üssünü alır. Sonra, sonucu tekrar ikilik tabana çevirir ve ikiliyi dizgeye çevirerek döndürür. Bu fonksiyon, Feistel ağında her turda sağ yarısını karıştırmak için kullanılır.

3. ALGORİTMA MODELİ



Algoritma Şifreleme işlemini gerçekleştirirken Feistel Mimarisini kullanır. Mimari de kullanılan F fonksiyonu daha önceden tanımını verdiğimiz scramble fonksiyonudur.



Bu diyagramda ise algoritmanın şifre çözme işlemini nasıl yaptığı gösterilmiştir.

4. ALGORİTMANIN ÇALIŞMA ÇIKTILARI

Program çalıştırıldığında Şekil 4.1.‘de görüldüğü gibi bir ekran açılmaktadır. Bu ekranda;

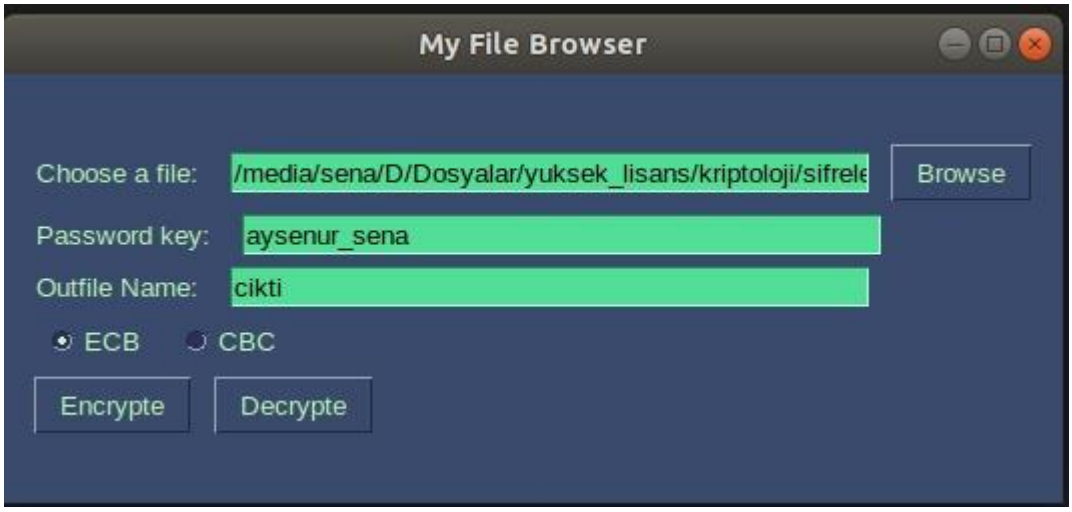
Browse butonu ile şifrelenmek istenen txt uzantılı dosyanın yolu programa tanıtılır.

Password Key bölümüne şifreleme ve deşifreleme işleminde kullanılacak olan anahtar değeri girilir.

Outfile Name bölümünde ise şifreleme ve deşifreleme işlemi tamamlandıktan sonra elde edilen sonucun yazılacağı txt dosyasının adı belirlenir.

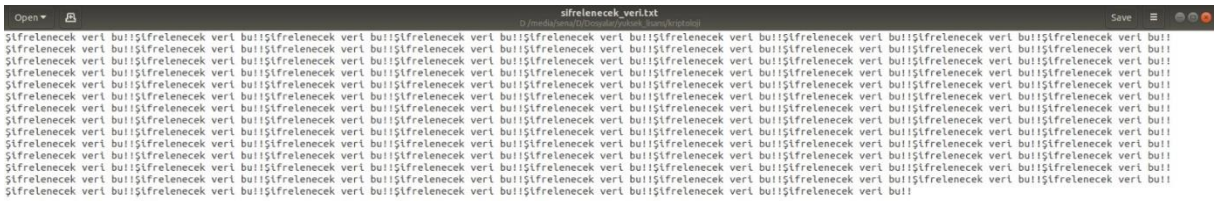
Daha sonra ECB veya CBC modlarından istenen şifreleme modu seçilir.

Son kısımda ise şifreleme ve şifre çözme işlemlerinden hangisi yapılacak ise o butona basılarak işlem başlatılır.

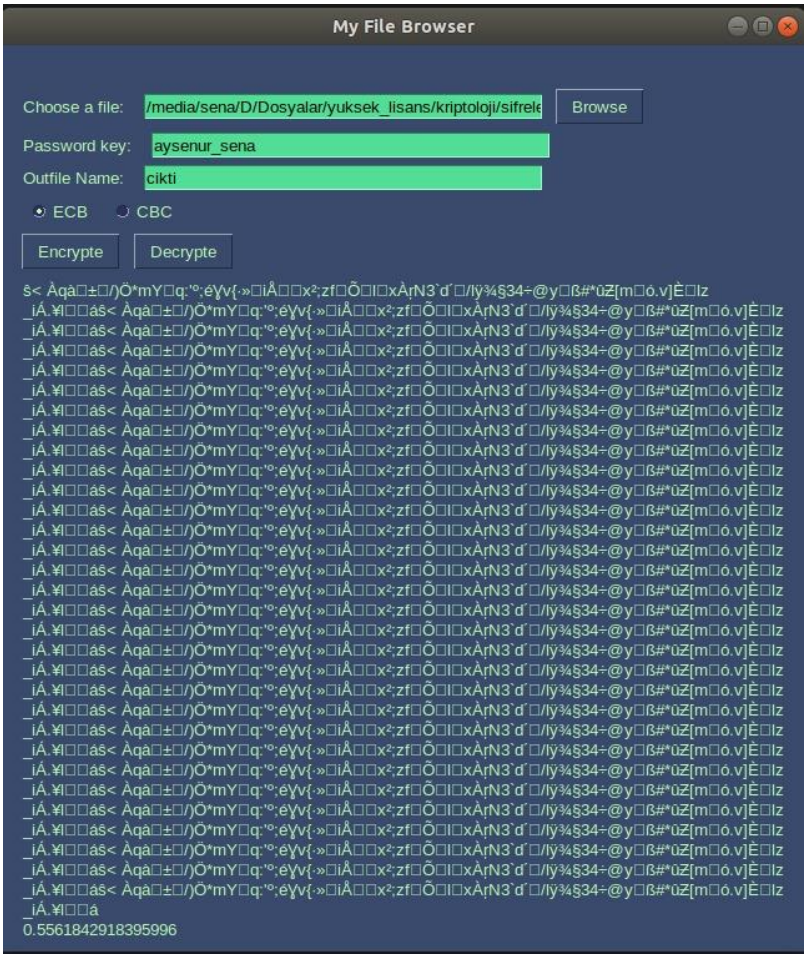


Şekil 4.1. Başlangıç Ekranı.

Şekil 4.2. ‘ de görülen metin algoritmaya verilmiş ve ECB modunda şifreleme yapılmıştır. Elde edilen şifrelenmiş metin Şekil 4.3.’ de gösterilmiştir.



Şekil 4.2. Şifrelenecek Metin.



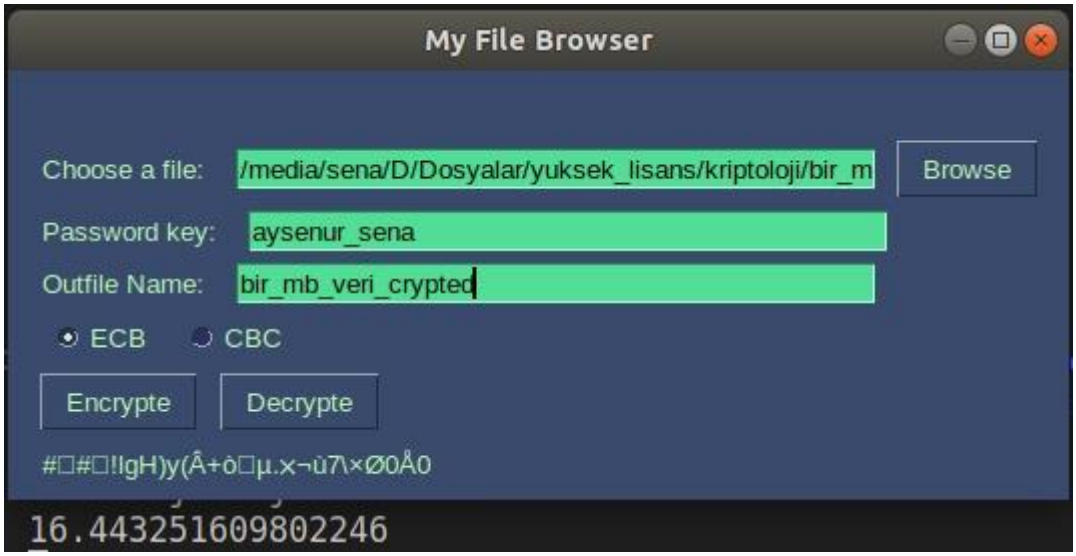
Şekil 4.3. Şifrelenmiş Metin.

Aynı adımlar uygulanarak şifre çözme işlemi de gerçekleştirilir. Şifre çözmede şifrelenmiş metin dosyası programa yüklenir ve şifreleme yapılırken kullanılan mod seçilerek Decrypte butonuna basılır. Şekil 4.4.' de görüldüğü gibi.

Şekil 4.4. Şifre Çözme.

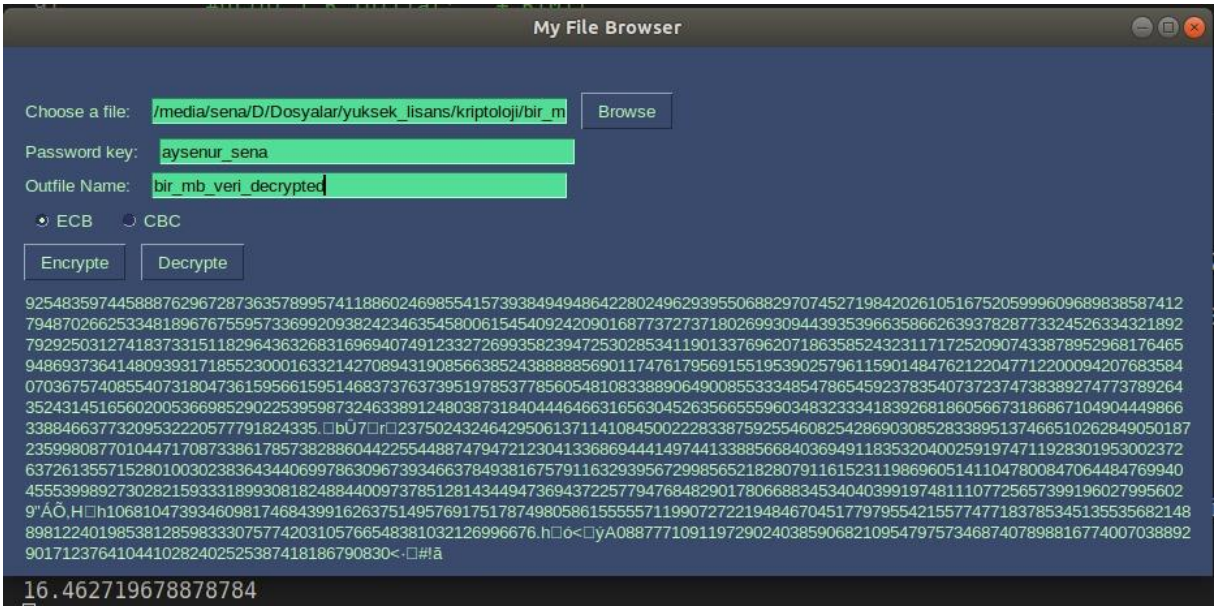
5. ALGORİTMANIN PERFORMANS SONUÇLARI

Algoritmanın çalışma performansı, 1 MB boyutundaki veriyi şifreleme süresi hesaplanarak bulunmuştur. Şekil 5.1. 'de görüldüğü gibi 1 MB boyutundaki açık metin dosyası algoritmaya verilmiş ve şifreleme işlemi yaklaşık 16 saniye sürmüştür.



Şekil 5.1. 1 MB Boyutundaki Dosyanın Şifrelenmesi.

Şifrelenen dosyanın şifre çözme işlemi de şifrelemeye benzer olarak yaklaşık 16 saniye sürmüştür. Şekil 5.2.'de şifre çözme işleminin çıktılarını görebiliriz.



Şekil 5.2. Şifre Çözme İşleminin Çalışma Süresi.