T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BSM 465 KRİPTOLOJİYE GİRİŞ DERSİ PROJE RAPORU

Şifreleme Algoritması Tasarımı

HAZIRLAYANLAR

G201210036- Ayşe Esra AŞCI
B201210066- Hüsna ALTIN

Dersi Veren: Ünal ÇAVUŞOĞLU

Projeye ait GitHub Linki:https://github.com/husnaaltin/Kriptoloji

2023-2024 Güz Dönemi

Verilen proje ödevi kapsamında bizden güvenli iletişim ve veri koruma amaçlarıyla kullanılmak üzere yeni bir şifreleme algoritması tasarlamamız istenmiş olup projenin, temel şifreleme prensiplerini kullanarak bir blok şifreleme algoritması tasarımı ve performans ölçümleri üzerine odaklanması talep edilmiştir. Literatürdeki algoritmalardan esinlenerek yeni bir algoritma tasarımı yapmamız beklenmektedir. Ödevde istenen kriterler göz önüne alınarak proje oluşturulmuş olup proje oluşturulurken izlenen adımlar aşağıdaki gibidir.

1.Literatür Taraması

Ödeve başlarken ilk olarak fikir edinmek ve bilgi toplamak adına literatür taraması yapılmıştır. Şu anda ve geçmişte kullanılan popüler şifreleme algoritmaları araştırılmıştır. AES (Advanced Encryption Standard), DES (Data Encryption Standard), 3DES (Triple DES), RSA (Rivest-Shamir-Adleman), Diffie-Hellman ve Caesar gibi şifreleme yöntemleri göz önünde bulundurulmuş ve çalışma mantıkları incelenmiştir. Ayrıca bu şifreleme algoritmalarından ilham alınarak oluşturulmuş farklı üniversitelerde bulunan öğrencilerin projeleri incelenmiştir. Nurettin TOPALOĞLU, M. Hanefi CALP ve Burak TÜRK' e ait "Bilgi Güvenliği Kapsamında Yeni Bir Veri Şifreleme Algoritması Tasarımı ve Gerçekleştirilmesi" çalışması, Aysun COŞKUN ve Ülkü Ülker'e ait "Ulusal Bilgi Güvenliğine Yönelik Bir Kriptografi Algoritması Geliştirilmesi ve Harf Frekans Analizine Karşı Güvenirlik Tespiti" adlı çalışma proje için bilgi toplama ve ilham verme konusunda yardım alınan kaynaklardan bazılarıdır. Bu çalışmalar incelenmiş ve projenin gerçekleştirilebilmesi için gereken bilgiler elde edildikten sonra bir sonraki adıma geçilmiştir.

2. Temel Prensiplerin Belirlenmesi

Çalışmamızda gönderici ve alıcı arasındaki iletişim, simetrik şifreleme tekniği kullanılarak gerçekleşir. Bu şifreleme algoritmasında asimetrik şifrelemenin aksine ortak bir tek anahtar vardır. Kullanıcıdan alınan mesaj şifrelenirken pi sayısının rakamlarının belli bir düzene veya örüntüye göre gitmeyip kendini tekrar etmemesinin oluşturduğu karmaşıklıktan ve Fibonacci dizisinin uyumundan yararlanılmıştır. Katmanlı bir mimariye sahip olan algoritma için sabit bir anahtar boyutu tanımlanmamıştır.

Şifreleme algoritması her bir karakteri belirli bir düzen içinde şifreleyen bir basit şifreleme yöntemi kullanmaktadır. Şifreleme işlemi sırasında, her karakterin başındaki iki değer korunurken, karakterin ortasına, bir dinamik parça eklenmektedir. Son olarak, karakterin sonundaki iki değer de korunmaktadır. Buna karşılık, şifre çözme işlevi (simple_decryption fonksiyonu) her seferinde 8 karakterlik bir blok alır ve her bir karakteri çözmek için tersine bir algoritma uygulanmaktadır.

3. Algoritma Tasarımı

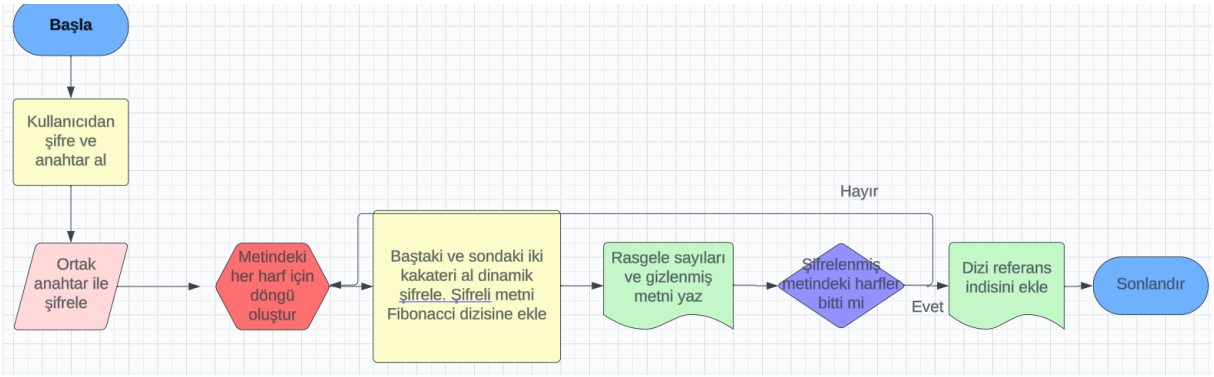
Algoritma tasarımı yaparken incelediğimiz kaynaklarda pi sayısı ve fibonacci sayı dizisinin kullanılması bize bir fikir sundu. Düşündüklerimizi yeni bir şifreleme algoritmasına dönüştürdük Şifreleme algoritması her bir karakteri belirli bir düzen içinde şifreleyen bir basit şifreleme yöntemi kullanmaktadır. Şifreleme işlemi sırasında, her karakterin başındaki iki değer korunurken, karakterin ortasına, bir dinamik parça eklenmektedir. Son olarak, karakterin sonundaki iki değer de korunmaktadır. Buna karşılık, şifre çözme işlevi (simple_decryption fonksiyonu) her seferinde 8 karakterlik bir blok alır ve her bir karakteri çözmek için tersine bir algoritma uygulanmaktadır.

Temelde iki ana fonksiyon olarak simple_encryption ve simple_decryption fonksiyonlarını kullandık. Şifreleme kısmı için kullanıcıdan girilen verinin baş ve son iki hanesi (char[:2] ve char[-2:]) alınarak parçalanıyor. Bu parçalar simple_algorithm fonksiyonu kullanılarak anahtar karakterleriyle işlenir ve şifrelenmiş karakterler elde edilir. Şifrelenmiş karakterler birleştirilerek şifrelenmiş veri oluşturulur.

Şifrenin çözülmesinde şifrelenmiş veri parçalara ayrılır ve her parça simple_algorithm_reverse fonksiyonu kullanılarak anahtar karakterleriyle işlenir. İşlenmiş parçalar birleştirilerek orijinal veri elde edilir. Şifreleme algoritmasında anahtar boyutu, kullanıcının girdiği encryption_key veya decryption_key olarak adlandırılan anahtarın uzunluğudur. Bu anahtar, simple_encryption ve simple_decryption metodları içinde kullanılır.

Bu döngü, kullanıcının girdiği veri (message) ile anahtarın her bir karakterini sırayla eşleştirir. Bu eşleştirme, anahtarın uzunluğu kadar devam eder.

Örneğin, eğer kullanıcı "12345" gibi bir beş karakterli bir anahtar girdiyse, algoritma her karakter için bu anahtarı kullanacak ve beş karakterlik bir anahtar boyutu olacaktır. Anahtarın her bir karakteri, verinin ilgili karakteri ile eşleştirilir ve şifreleme/çözme işlemi bu eşleştirmeler üzerinden yapılır.



4. Anahtar Üretimi

Anahtar üretim kısmında öncelikle kulanıcı, şifreleme ve çözme işlemleri için kullanılacak anahtar kelimeyi belirler. Bu kelime, şifreleme algoritmasının temelini oluşturur.

Kütüphaneler aracılığıyla pi sayısının değeri alınır ve bu sayı üzerinde matematiksel işlemler gerçekleştirilir. Aynı zamanda Fibonacci dizisi belirli bir uzunluğa kadar oluşturulur. Bu değerler, anahtar üretim sürecinde kullanılacak özel değerlerdir.

Anahtarın Hesaplanması:

Kullanıcının girmiş olduğu anahtarın her bir karakteri ile özel matematiksel işlemler gerçekleştirilir.

Bu işlemler sonucunda elde edilen değerler, anahtarın bir parçasını oluşturur. Rastgele bir sayı eklenerek güvenlik artırılır. Hesaplanan değerler, SHA-256 hash fonksiyonu ile özetlenir.

Şifrelenmiş verinin çözülmesi için ters anahtar hesaplanır.

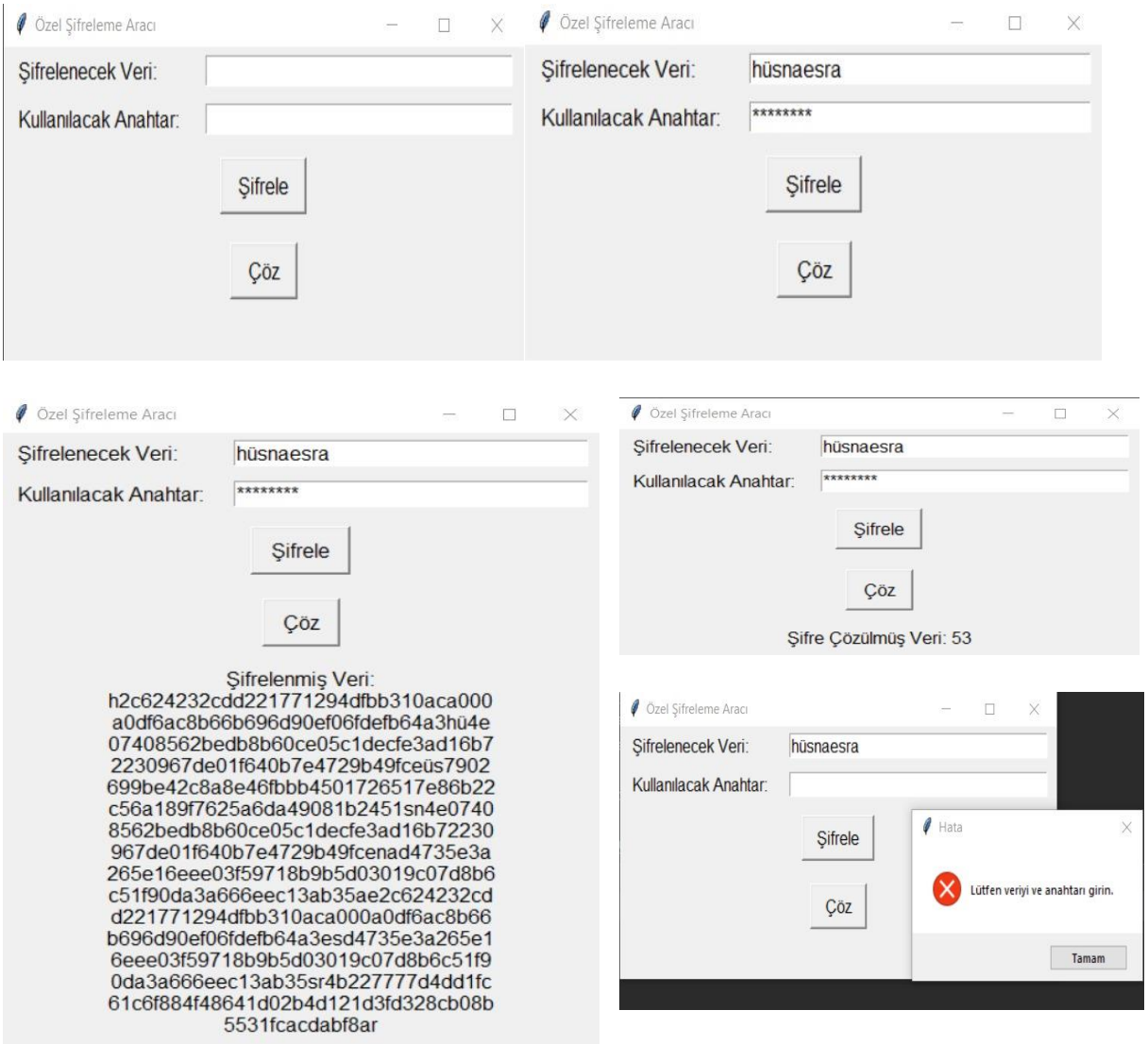
Şifrelenmiş verinin belirli bir bölümü ve kullanıcının girmiş olduğu anahtara karşılık gelen karakteri ile matematiksel işlemler gerçekleştirilir. Bu işlemler sonucunda orijinal anahtarın bir bölümü elde edilir. Şifrelenmiş mesajın bu bölümü üzerinden bir SHA-256 hash hesaplanır. Rastgele bir sayı çıkarılarak elde edilen değer üzerinden ters işlem yapılır.

Girilen anahtarın her karakteri ile çözülmüş mesajın karakterleri arasında matematiksel işlemler gerçekleştirilir.

Bu süreçler, kullanıcının belirlediği anahtar kelime ve matematiksel işlemlerin kombinasyonuyla özelleştirilmiş bir şifreleme algoritması oluşturur. Bu algoritma, verinin şifrelenmesi ve çözülmesi için kullanılır.

5. Algoritmanın Kodlanması

Proje Python dilinde PyCharm ortamında yazılmış olup şifrelenecek olan veri ve kullanılacak anahtarlar kullanıcıdan alınırken, şifreleme ve şifre çözme işlemi sonucu elde edilen sonuçlara ait ekran görüntüleri aşağıdaki gibidir.



Yukarıdaki görsellerde de görüldüğü gibi kullanıcıdan bir metin ve anahtar alınmakta şifrele butonuna basıldığında metin şifrelenerek şifrelenmiş veri ekrana sonuç olarak gelir. Daha sonra çöz butonuna basıldığında ise çözülen veri ekrana çıkar. Anahtar ve şifrelenecek metin alanı ise boş geçilememektedir.