T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BSM 465 KRİPTOLOJİYE GİRİŞ

PROJESİ

B211210383 Yiğit Çuhadar

Bölüm : BİLGİSAYAR MÜHENDİSLİĞİ
Öğretmen : Doç.Dr. ÜNAL ÇAVUŞOĞLU

2023-2024 GÜZ DÖNEMİ

İÇİNDEKİLER

1. Giriş

Bu rapor, kriptografi alanında geliştirilen bir şifreleme ve şifre çözme uygulamasının tasarımı ve uygulanması hakkında ayrıntılı bir açıklama sunmaktadır. Projenin amacı, güvenli ve etkili bir şifreleme algoritması geliştirmek ve bu algoritma ile metinleri şifreleyip çözmek için bir kullanıcı arayüzü sağlamaktır.

2. Temel Prensiplerin Belirlenmesi

2.1. Şifreleme Türü

Projede simetrik şifreleme kullanılması kararı alındı. Bu, daha hızlı işlemler ve daha basit anahtar yönetimi sağlar. Şifreleme ve şifre çözme işlemleri için aynı anahtar kullanılır.

2.2. Blok Boyutu ve Anahtar Boyutu

Şifreleme için 128-bit blok boyutu ve 256-bit anahtar boyutu seçildi. Bu, güçlü şifreleme ve yüksek güvenlik seviyesi sağlamak için önemlidir. Anahtarın uzunluğu, güvenlik açısından kritik bir parametredir.

2.3. Şifreleme Modu

CBC (Cipher Block Chaining) modu tercih edildi. Her bloğun şifrelenmesi bir önceki bloğa bağlı olduğu için tekrar saldırılarına karşı daha dirençli bir yapı sağlar. CBC modu, her bloğun önceki şifreli blokla XOR işlemi uygulanarak şifrelendiği bir moddur.

2.4. Döngü Sayısı

Projede, şifreleme ve şifre çözme işlemlerinde 10 döngü kullanıldı. Bu döngüler, anahtarın daha fazla karıştırılmasını ve güvenliğin artırılmasını sağlar. Döngü sayısı, güvenlik ve performans dengesi gözetilerek seçildi.

3. Algoritma Tasarımı

Projenin algoritma tasarımı şu şekildedir:

3.1. Anahtar Üretimi

Güvenli anahtar üretimi için Python'un **Crypto** kütüphanesindeki **get_random_bytes()** fonksiyonu kullanıldı. Bu fonksiyon, belirtilen uzunlukta rastgele bir anahtar üretir. Projede 256-bit (32 byte) uzunluğunda anahtarlar kullanıldı.

3.2. Şifreleme İşlemi

1. **Metin Hazırlama**: Kullanıcıdan alınan metin, 128-bit (16 byte) bloklara ayrılır. Blok boyutu, AES şifreleme standardının bir gereksinimidir.
2. **Başlangıç Vektörü (IV)**: Her şifreleme işlemi için rastgele bir başlangıç vektörü (IV) oluşturulur. IV, şifreleme işleminin her seferinde farklı olmasını sağlar ve güvenliği artırır.
3. **Blok Şifreleme**: Her blok, AES (Advanced Encryption Standard) algoritması kullanılarak anahtar ile birlikte şifrelenir. Şifreleme modu olarak CBC kullanıldığı için her bloğun önceki şifreli blokla XOR işlemi uygulanarak şifrelenir.

3.3. Şifre Çözme İşlemi

1. **Şifreli Metni Bloklara Ayırma**: Şifreli metin, 128-bit (16 byte) bloklara ayrılır.
2. **Blok Şifre Çözme**: Her blok, AES algoritması kullanılarak anahtar ile birlikte şifre çözülür. CBC modu gereği her bloğun önceki şifreli blokla XOR işlemi uygulanarak şifre çözülür.

3.4. Anahtar Üretimi

Güvenli anahtar üretimi için **get_random_bytes()** fonksiyonu kullanıldı. Bu fonksiyon, kriptografik güvenliğe sahip rastgele veriler üretir ve anahtarların tahmin edilmesini zorlaştırır.

4. Uygulama ve Performans Ölçümü

Projeyi kullanıcılar, bir grafik arayüz üzerinden metin şifreleme ve çözme işlemlerini gerçekleştirebilirler. Ayrıca, projede şifreleme süresi, şifre çözme süresi ve bellek kullanımı ölçümleri yapıldı.

- **Şifreleme Süresi**: Metin şifreleme işleminin ne kadar sürede tamamlandığını ölçer. Genellikle milisaniye cinsinden ölçülür.
- **Şifre Çözme Süresi**: Şifreli metin çözme işleminin ne kadar sürede tamamlandığını ölçer. Genellikle milisaniye cinsinden ölçülür.
- **Bellek Kullanımı**: Uygulamanın çalıştığı sırada ne kadar bellek kullandığını ölçer. Genellikle megabayt (MB) cinsinden ölçülür.

5. Sonuçlar

Bu proje, güvenli bir simetrik şifreleme algoritması tasarlayarak ve bunu bir kullanıcı arayüzü ile entegre ederek başarılı bir şekilde tamamlandı. Proje sonucunda, güvenli veri iletimi ve saklaması için kullanılabilir bir şifreleme uygulaması oluşturuldu.