**T.C.**

**KIRKLARELİ ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**



**YAZ16402 YAZILIM GÜVENLİĞİ**

**Bütünleme Ödevi**

**GRUP ÜYELERİ**

**1180505612 – Mekan MYRADOV**

**PROJENİN GİTHUB LİNKİ**

<https://github.com/MekanMyradov/RSA_simple_app>

**KIRKLARELİ, 2022**

İçindekiler

[**1** **Giriş** 3](#_Toc107061305)

[**2** **Asimetrik Şifreleme Algoritmaları ve RSA** 3](#_Toc107061306)

[**3** **Bir Mesajı Şifreleyen ve Deşifre Eden Basit RSA Uygulaması** 3](#_Toc107061307)

[**4** **Kaynakça** 5](#_Toc107061308)

# **Giriş**

Teknolojinin gelişmesiyle birlikte veri akışının hızı da artmıştır. Bu artış birçok avantajın yanında önemli miktarda riskler de oluşturmuştur. A noktasından B noktasına giderken verilerin bütünlüğünün bozulması veya yetkisiz kişiler tarafından ele geçirilmesi bu risklere örnek olarak verilebilir. Bu tür risklerden kaçınmak için son 50 yılda çeşitli veri şifreleme algoritmaları geliştirilmiştir. Bu çalışmada asimetrik bir şifreleme algoritması olan RSA'dan bahsedilmiştir ve bu algoritma ile bir mesajı şifreleyen ve deşifre eden basit bir uygulamaya yer verilmiştir.

# **Asimetrik Şifreleme Algoritmaları ve RSA**

Asimetrik şifreleme algoritmaları, verileri şifrelemek ve şifresini çözmek için bir açık anahtar (public key) ve bir özel anahtar (private key) olmak üzere iki anahtar gerektirir. Bu tür algoritmalar kullanıldığında, açık anahtarla şifrelenen verilerin şifresi özel anahtarla çözülebilir. Günümüzde en yaygın olarak kullanılan asimetrik şifreleme algoritması RSA algoritmasıdır.

RSA, 1977'de üç bilim adamı, **Ron Rivest**, **Adi Shamir** ve **Leonard Adleman** tarafından geliştirilmiştir [1]. Algoritmanın adı, algoritmayı geliştiren kişiler tarafından yazılan bilimsel makaledeki soyadlarının ilk harflerinin birleşiminden gelmektedir [2].

RSA algoritması veriyi aşağıdaki formülü kullanarak şifrelemektedir. Bu formülde **P** veri, **C** verinin şifrelenmiş hali, **e** açık anahtar, **d** özel anahtar ve **n** iki büyük asal sayının çarpımıdır [3].

RSA, şifrelenen veriyi deşifre etmek için ise bu formülü kullanmaktadır:

# **Bir Mesajı Şifreleyen ve Deşifre Eden Basit RSA Uygulaması**

Geliştirilecek uygulama Python **PyCryptodome** kütüphanesini kullanmaktadır. **PyCryptodome** kütüphanesinin kurulumu **Şekil 1**’de gösterilmektedir.

Text

Description automatically generated

**Şekil 1:** PyCryptodome kütüphanesinin kurulumu

Kütüphane kurulduktan sonra kod dosyasına dahil edilir:

from Crypto.PublicKey import RSA

from Crypto.Cipher import PKCS1\_OAEP

from Crypto import Random

1024 bit uzunluğunda açık ve özel anahtarlar oluşturulur:

random\_generator = Random.new().read

private\_key = RSA.generate(1024, random\_generator)

public\_key = private\_key.publickey()

Şifrelenecek cümle belirlenir:

plain\_text = "Basit uygulama."

print("Şifrelenecek cümle:")

print(plain\_text)

print("")

Açık anahtar ile bir cipher oluşturularak şifreleme işlemi yapılır:

cipher = PKCS1\_OAEP.new(public\_key)

encrypted\_text = cipher.encrypt(plain\_text.encode("utf-8"))

print("Şifrelenmiş hali:")

print(encrypted\_text)

print("")

Özel anahtar ile yeni bir cipher oluşturularak şifrelenen veri deşifre edilir:

cipher = PKCS1\_OAEP.new(private\_key)

decrypted\_text = cipher.decrypt(encrypted\_text).decode("utf-8")

print("Deşifre edilmiş hali:")

print(decrypted\_text)

Yukarıda verilen kod parçaları birleştirilip çalıştırıldığında uygulamanın sonucu **Şekil 2**'deki gibi olacaktır.

Text

Description automatically generated

**Şekil 2:** Uygulamanın sonucu

# **Kaynakça**

|  |  |
| --- | --- |
| [1] | «Wikipedia,» [Çevrimiçi]. Available: https://en.wikipedia.org/wiki/RSA\_(cryptosystem). [Erişildi: 25 Haziran 2022]. |
| [2] | «MIT CSAIL,» [Çevrimiçi]. Available: http://people.csail.mit.edu/rivest/Rsapaper.pdf. [Erişildi: 25 Haziran 2022]. |
| [3] | «Study,» [Çevrimiçi]. Available: https://study.com/academy/lesson/modern-cryptographic-algorithms-function-effectiveness.html. [Erişildi: 25 Haziran 2022]. |