# BİL 4106 Veri Güvenliği – Ödev

Dersin Öğr. Elemanı: Dr. Öğr. Üyesi Mete Eminağaoğlu

**Ödev Konusu:** Öğrenciler, aşağıda kısaca açıklanan 2 farklı kriptografi konusundan bir tanesini seçip kodlayacaktır. Aşağıda çok kısa özet bilgiler verilmiştir. Detaylı açıklamalar, sadece bir kere, ders saatinde yapılacaktır. O derse gelemeyen, vb öğrencilerin ödevi yanlış / eksik vb anladığı için yetersiz bir ödev iletmesinden sadece o öğrencinin kendisi sorumludur.

---

**Ödevin Son Teslim Tarihi:** 12 Mayıs 2019, 23:00 (TSİ)

**Ödevin Sunumu:** 13 Mayıs 2019, 09:30

***DİKKAT! Ödevlerin teslimi yeterli değildir. 13 Mayıs 2019 günü ders saatinde tüm öğrencilerin derse gelmesi ve ödevlerini sınıfta sunmaları zorunludur.*** *Ödevini zamanında teslim etmiş olsa bile,13 Mayıs 2019 tarihindeki* ***derse gelmeyen ve sunumu yapmayan öğrenciler de ödevden 0 (sıfır) alacaktır.***

**Ödevin Teslim Şekli:**

CSC ÖBS (Moodle) sistemindeki ders sayfasında açılacak olan ödev yükleme (assignment) alanına; tüm dosyalar, rapor, vb. **zip / rar sıkıştırılmış tek bir dosya olarak yüklenecektir**.

---

Bu ödev **tek kişi ya da iki (2) kişilik ekipler şeklinde yapılabilir.**

**Ödev Konusu - 1:**

Açık Anahtar Altyapısı (PKI) kullanan, Türkçe arayüzlü bir veri (dosya) şifreleme / imzalama uygulaması.

- Masa üstü, web tabanlı, veya mobil uygulama olabilir. Öğrencinin tercihine bırakılmıştır.
- Hash, simetrik şifreleme ve asimetrik şifreleme işlemlerinin hepsi gerekli yerlerde kullanılacaktır.
- Kullanıcıların (A ve B) henüz asimetrik anahtarları (private ve public) yok ise, önce o kullanıcı için bu anahtarları oluşturulmalıdır.
- A Kullanıcısı, ekrandaki menüden dosya seçecek ve ekrandaki seçenekleri kullanarak;
    - Ya, dosyayı sadece imzalayacak,
    - Ya, dosyayı sadece şifreleyecek,
    - Ya da, dosyayı hem imzalayacak, hem de şifreleyecektir.

- B Kullanıcısı, ekrandaki menüden kendisine A'dan iletilen dosyayı seçecek ve ekranda **kendisine uygulama tarafından sunulan yönlendirmeleri** kullanarak;
    - Ya, dosyayı açıp A'dan geldiğini teyit ettirecek yani imzayı doğrulayacak (validate signature),
    - Ya, dosya sadece şifrendi ise, deşifreleyip dosyayı açıp okuyabilecek,
    - Ya da, A dosyayı hem imzalamış hem şifrelemişse, dosyayı hem deşifreleyip açıp okuyabilecek, A'dan geldiğini teyit ettirecek yani imzayı doğrulayacak.

- Hazır fonksiyon, kütüphane, vb. kullanımı serbesttir. Özellikle güncel endüstri standardı algoritmaları hazır ve güvenilir kaynaklardan kullanmanız önerilir (SHA-256, SHA-512, RSA, ECC, AES, vb). Yani, bu algoritmaları baştan sizin yazmanız istenmemektedir.
- Kodlama kısmında, **bu programlama dillerinden bir tanesini kullanabilirsiniz: C, C++, C#, .Net, Java, Python.**

**Ödevde Teslim Edilecekler:**

1-Programın tüm kaynak kodları, bağlantılı kütüphane, dizinler, vb.

2-Kullanılan algoritmalar, vb ile ilgili kısa bilgiler / notlar (istenirse kaynak kod içine de açıklamalar olarak eklenebilir).

**Ödev Konusu - 2:**

Tamamen kendi tasarlayacağınız şekilde, Türkçe arayüzlü bir kriptografik hash uygulaması.

- Masa üstü, web tabanlı, veya mobil uygulama olabilir. Öğrencinin tercihine bırakılmıştır.
- **Hazır fonksiyon, kütüphane, hazır kriptografik algoritma, araç, vb kullanımı yasaktır. Sizin kendi hash fonksiyonunuzu tasarlayıp kodlamanız gerekmektedir.**
- Kullanıcı, ekrandaki menüden bir dosya seçecek ve sizin özgün hash algoritması ile hash değerini alması ve ekrana bu değeri yazması sağlanacaktır.
- Dosya tipi herhangi bir şey olabilir (binary, Ascii, doc, pdf, txt, xml, html, resim / video formatları, vb)
- Hash çıktısı (message digest / mesaj özütü) en az 256 bit olmalıdır.
- Dosya uzunluğu 250 MegaByte'ı geçiyor ise, dosyayı açıp hash işlemine geçmeden önce, dosyayı parçalara ayırıp ayrı ayrı hash işlemine sokabilir. Ya da, daha zor bir yöntem olan dosya uzunluğu ne olursa olsun (gerekirse 40 Gigabyte, 1 TeraByte, vb), dosyanın tamamını açıp hash işlemine geçecek çözüm bulmak isteyenler bunu da tercih edebilir.
- Özgün hash algoritma ve fonksiyonlarınızda, XOR, OR, AND, NOT, (right / left) circular shift k bits, addition mod $2^n$ kullanılması gereklidir. Nerede, ne şekilde bunları kullanacağınız ve hash işlemlerinin kaç tur olacağı, sabit vektör blokları kullanıp kullanmayacağınız size kalmıştır.
- Kodlama kısmında, **bu programlama dillerinden bir tanesini kullanabilirsiniz: C, C++, C#, .Net, Java, Python.**

**Ödevde Teslim Edilecekler:**

1-Programın tüm kaynak kodları, bağlantılı kütüphane, dizinler, vb.

2-Kullanılan algoritmalar, vb ile ilgili kısa bilgiler / notlar (istenirse kaynak kod içine de açıklamalar olarak eklenebilir).