

425492 Alazttin Uysel  
449821 Gagatay Turunc  
449823 Ozkan Teber

BIL3023 Bilgi Güvenliği  
Proje Raporu

## 1. Giriş

Bu proje kapsamında, Java programlama dili ve ~~socket~~ socket programlama teknikleri kullanılarak güvenli bir mesajlaşma sistemi tasarlanmıştır. Geliştirildiğimiz program, kullanıcı ad/parsel doğrulaması yerin, parolanın bir görsel içeriğine LSB Steganografi yöntemiyle gizleyerek bir doğrulama mekanizması sunar. Ayrıca, mesajlaşım trafiği DES algoritması ile şifrelenerek, sunucu üzerinde güvenli bir şekilde iletilmektedir.

Proje, istenci - sunucu mimarisini üzerine inşa edilmiş olup, gelen istenci destek ve gelenimdisi mesajlaşmaya özelligile genisuz mesajlaşmaya sistemlerine benzer yapıdadır.

## 2. Sistem Tasarımı

Bu bölümde, projenin isterlerini karşılayan on temel fonksiyonun teknik detayları ve algoritmik arayışları açıklanmıştır.

### 2.1. Gelen İstenci Destek:

Sunucu mimarisini, aynı anda birden birden fazla istemcinin bağlanabilmesi üzerine kurulmuştur. Bu yapı için Java'nın ServerSocket sınıfı ile 5555 portu dinlenmeye alınmış ve Multi-threading yapı kullanılmıştır.

Gereklesini: Ana sunucu döngüsü (while(true)), her yeni bağlantı isteği (accept) geldiğinde, bu bağlantıyı yönetmek için başımsız galiba bir ClientHandler iş parçası oluşturur.

• Sunucu: Bu seyede, örneğin Özkan, Gagatay ve Alazttin aynı anda sunucuya bağlanıp işlem yapabilselerin birbirlerinin veri trafiğini veya sunucunun genel işleyisini bloke etmezler.

## 2.2 Grafik Arayüz ve Kayıt İşlemi

Kullanıcı deneyimini bolystastirmak amacıyla Java Swing kütüphanesi kullanılarak bir kayıt formu (Registersform) tasarlanmıştır.

- Girdi Yontemi: Kullanıcıdan "Kullanıcı Adı" (codename), "Parola" (secretkey) metin olarak alınır.
- Görüsel Seçim: Steganografi işleminin uygulanabilmesi için kayipsiz sıkıştırma sunan PNG formatındaki dosyaların seçilmesine izin veren JFileChooser entegre edilmiştir. Bu arayüz güvenlik parametresinin kullanıcı dostu bir şekilde sisteme girilmesini sağlar.

## 2.3 Steganografi: LSB ile Anahat Gizleme

Projenin en kritik güvenlik katmanlarından biri olan ve: gizlene işlemi, LSB algoritması ile istemci istemci tarafından gerçekleştirilir. Anlaş, parolanın üzerindeki açık metin olarak dolgusunu engellerken.

- Algoritma: Seçilen resmin pikselleri, BufferedImage sınıfı ile taranır. Her bir pikselin RGB değerinin son biti, bit düzeyinde (bitwise) işlemle değiştirilir.
- AND operatörü ( $\text{pixel} \& \text{0xFFFFFFFF}$ ) ile son bit sifirlanır.
- OR operatörü ( $| 1 \text{ bit}$ ) ile parolanın ilgili biti piksele yazılır.
- Güvenlik Artırımı: Pikseller sırayla değil, Collections.shuffle kullanılarak karıştırılmış bir sırayla seçilir. Bu sayede jörsel olabileceği desen bozulmalarının önüne geçilir.

## 2.4 Sunucu Tarafında Anahat Çıkarma

İstemci tarafından gönderilen "Stego-Resim" sunuya ulaşlığında, sunucu bu resmi işleyerek içindeki gizli anahatı ayırtırır.

- Sürec: Sunucu sifrelene aşamasında kullanılan tüm değerler ile aynı piksel sırasını üretir. Her pikselin son biti ( $\text{pixel} \& 1$ ) okunarak bir grupta toplanır ve 8 bitlik gruplar halinde karakterlere dönüştürülür.

- Eşlesme: Çıkarılan anahat, kullanıcının sisteme girdiği kullanıcı adıyla eşleştirilerek sunucu belleğindeki (ConcurrentHashMap) güvenli kasada saklanır.

## 2.5. Aktif Kullanıcı Listesi ve Durum Yönetimi:

Sunucu, sisteme bağlı olan kullanıcıları enlik olarak takip eder.

•**Canlı Liste**: activeClients isimli Thread-safe bir harita yapısı kullanılarak online kullanıcılar hafızada tutulur.

•**Görselleştirme**: Sunucu, periyodik olarak veya kullanıcı listesinde değişim olduğunda, tüm istencilerse güncel listeki gönderir. İstemi arayüzünde sunucudan gelen ve kişiye göre aktif kullanıcılar Yeşil, çevrimdışı kullanıcılar kirmizi renk ile dinamik olarak gösterilir.

## 2.6. Çevrimdışı (Offline) Mesajlaşma Mimarisi:

Sistemdeki o en sisteme bağlı olmasa bile mesajın kaybolmasına sağlayan "Depo ve İlet" modelini uygular.

•**Mekanizma**: Bir istemi mesaj gönderdiğinde, sunucu hedef kullanıcının activeClients listesinde olup olmadığını bakar.

•**Depolama**: Eğer hedef kullanıcı çevrimdışı ise, mesaj o kullanıcıya özel ayrılmış bir mesaj kuyruğuna (offline Messages Map yapısı) eklenir ve RAM üzerinde bekletilir.

## 2.7. Çevrimiçi Olunca Mesajları İletme

Kullanıcı sisteme giriş yapıp kimlik doğrulamasını tamamladığı anda sunucu bu kullanıcı adına bekleyen mesaj olup olmadığını kontrol eder.

•**Mesaj İletimi**: handleRegister metodu tetiklendiğinde, eğer kuyrukta bekleyen mesajlar varsa, bu mesajlar straight kullanıcının soket hattına yazılır. Mesajlar iletildiğten sonra sunucu kuyruğundan silinerek veri tekrarı önlenir.

## 2.8. İstemci Tarafında DES Şifrelene

İstemci, mesajı ağ kablosuna bürkmeden önce kendisi belirlediği anahtarla şifreler.

- Kriptografi: Java Kriptografi Mimarisi kullanılarak DES algoritması uygulanır. Blok şifrelene modu olarak ECB ve dolgu yöntemi olarak PKCS5 Padding tercih edilmiştir.

- İşlem: Kullanıcının resme gittiği enstitü (secret key), mesaj metnini şifreli bir 'bagıt' düzinine dönüştürmek için kullanılır.

## 2.9. Şifreli Mesajın İletimi

Ağ trafiği üzerinde hiçbir zaman açık metin veri dolaşmasın.

- Protokol: İstemci, şifrelediği wayi SENDI HEDEF-KULLANICI I ŞİFRELİ-VERI formatında paketleyerek sunucuya ileter. Bu paketleme seyesinde sunucu, mesajın içeriğini görmeden kimden geldiğini ve kimse gideceğini protokol başlığında zıtlar.

## 3.10. Sunucu Tarafında Desifrelene ve Yönlendirme

Sunucu, bir kopyu görevi görür.

- Desifrelene: Sunucu, gelen paketin kimden geldiğini bildiğince içinden kisinin kayıtlı ekranında resimden akardığı anahtarları kullanarak şifreli mesajı çözür ve açık metni elde eder.

- Veri Dağılımı: Mesajın alıcısı farklı bir anahtara sahip olduğu için sunucu elde ettiği açık metni bu kez alıcının anahtarı ile tekrar şifreler. Böylece mesaj, göndericiden alıcıya güvenli bir dönüşüm geçirerek iletilmiş olur.

### 3. Senaryo Analizleri

#### 3.1. Es zamanlı Geçimici İletişim (Online-Online)

Kullanıcı A (gönderici) ve kullanıcı B (alıcı) aynı anda sistem bağlıdır.

Aks: A mesajı gönderir → istemci A mesajı kendi anahtarıyla şifreler → Sunucu gösterir ve B'nin anahtarıyla tekrar şifreler → Kullanıcı B mesajı alır ve anahtarıyla çözür.

#### 3.2 Asenkron İletişim ve Ver. Saklama

Kullanıcı A girmişi, Kullanıcı B girmişi

Aks: Kullanıcı A mesajı gönderir. Sunucu, B'nin aktif olmadığını tespit eder.

Mesajı, B'nin kuyruğuna ekler. B sisteme giriş yapınca, sunucu kuyruktaki mesajları, B'ye ileter.

#### 3.3 Oturum Bazlı Anahtar Değişimi

Kullanıcı B, sunucuda bekleyen mesajları verken, sisteme farklı görsel/parot ile sisteme giriş yapar.

Aks: Sistem B'nin yeni anahtarını kabul eder ve iletişime izin verir. Ancak sunucuda bekleyen eski mesajlar eski anahtar ile şifrelenmişti. İzin B bu eski mesajları açarken şifre çözme hatası alır.

Bu durum hata olarak değerlendirilmelidir. Bir saldırgan B'nin kullanıcı adını ele geçirse dahi, B'nin orijinal görseline sahip olmadan geçmiş mesajlarını okuyamayacağını garanti eder.

## 4. Sistem Test Logları

Geliştirildiğimiz program çalışma zamanında ürettiği log kayıtları, incelenerek, algoritmaların doğru çalıştığı esasındaki veilerle doğrulanmıştır.

### 4.1 Steganografi (LSB) İşlen Kesti

StegoManager sınıfı tarafından resim işlenirken stegano-debug.txt dosyasında kaydedilen piksel değişim sıraları aşağıdaki gibidir.

[14:20:15] -- [ENCODE BAŞLATIL] key: mySecretKey -- [14:20:15] [DEĞİŞİM]  
Piksel(12,45) | Psk: LSB:0 → Yeni: LSB 1 | Girişler Bit 1 [14:20:15] [DEĞİŞİM] --

### 4.2 DES Analizi:

[CLIENT LOG] Kullanıcı Girişisi: "Mehmet" [CLIENT LOG] Des ile şifrelenen: "yT5+dfggKlo= [Network] Gönderilen paket: SEND| otken| yT5+dfggKlo=

[SERVER LOG] (Gelen şifreli (çözülmüş)): "yT5+dfggKlo=" [SERVER LOG] (Plain text): "Mehmet" [SERVER LOG]: (Ortaklıkın şifrelenisi): "Ab3Kp92msk="

### 4.3 Offline

[SERVER LOG]: (otken) kontr oluyor. [SERVER LOG] [STOK] otken offline. Mesaj kurye alındı. (otken sisteme giriş yapmış.) [SERVER LOG] [KAYIT] otken (Anahat: otken123) [SERVER LOG] otken için bekleyen mesajlar iletildi.

## 5. Sonuç

Bu proje kapsamında, Java teknolojileri kullanılarak Steganografi ve kriptografi tekniklerini birlestiren, güvenli bir enlit mesajlaşma sistemi başarıyla geliştirilmiştir.

429 492 Alattin Uysal  
449 821 Güray Turan  
449 823 Örken Teber