

**COMPUTER SYSTEM SECURİTY DERSİ**

**2.ÖDEV RAPORU**

AHMET İHSAN SAVAŞ

14121221028

**1.PROJENİN AMACI**

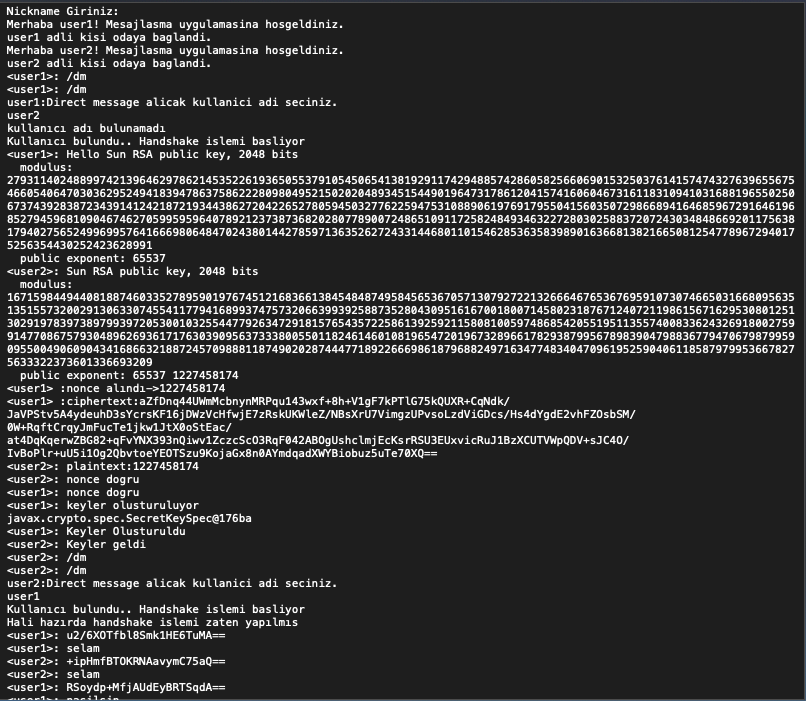
Bu projemizde bizden istenen derste öğrenmeye çalışmış olduğumuz SSL şifreleme yöntemini Socket.io sayesinde yapılan Client Server arasında ki mesajlaşma uygulamasına uyarlamak ve önce ki ödevde hazır olarak kullanmış olduğumuz RSA , AES , DES vb. şifreleme tekniklerini uygulamaya dökmemiz istenmektedir. Bu doğrultuda bizden Java programlama dilinde istenenleri yapmaya çalıştım.

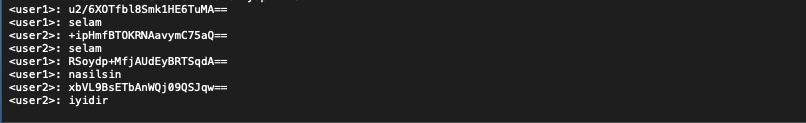
Soket Programlama kısmını internetten yardım alıp istenenler doğrultusunda uygulamaya entegre ettim. Soket programlama kısmı sadece herkesin ortak yayın yaptığı bir bölümdü , benim yaptığım kısımlar sonucunda kullanıcılar sadece özelden konuşarak birbirleriyle şifreli bir şekilde mesajlaşabiliyor .

**2.PROJE AÇIKLAMASI**

Projeye giriş yapıldığında ilk olarak “Server.java” dosyası daha sonra ise “Client.java” dosyası çalıştırılmaktadır . Sisteme birden fazla Client giriş yapacağı için “Client.java” birden fazla çalıştırılmaktadır. Bu java dosyaları çalıştırıldıktan sonra sistem Client tarafında bizden kullanıcı adı beklemektedir. Kullanıcı adı sisteme girildikten sonra birbirleriyle yapacağı görüşmeler için karşılıklı olarak sisteme “/dm” yazılarak birbirleriyle güvenli yoldan konuşabilmektedirler. Son olarak projedeki System.out.println() kısmının fazla olmasının nedeni bütün mesajları text dosyasına yazdırmam.

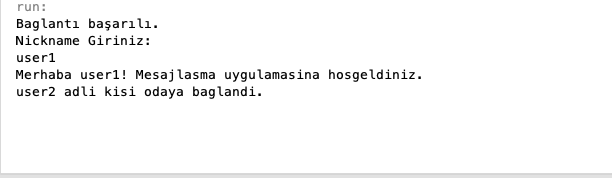
“Systemout.txt” dosyasına bütün mesajları yazdırdım, elde edilen görüntüler aşağıdadır.

****

****

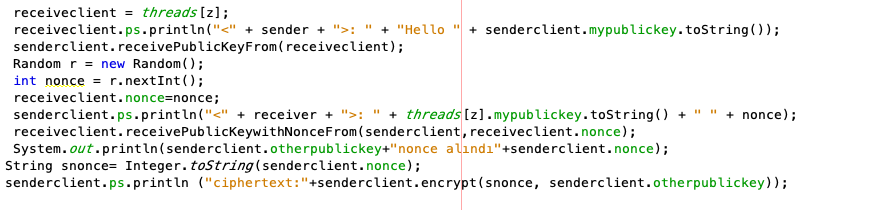
**3.PROJE AŞAMALARI**

**1.Handshake :** İlk olarak Client Server a bağlandığında Client tan bir kullanıcı adı istenmektedir.

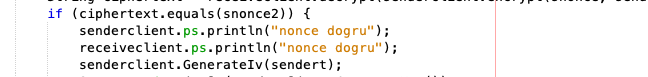


Kullanıcı adı User1 girildikten sonra diğer odaya bağlanan Client’ın kullanıcı adı User2 gözükmektedir. Daha sonra User1 /dm yazdıktan sonra sistem şifreli konuşulacak kullanıcı adı beklemektedir , User2 yazarak sadece User2 ile konuşulmak istenmektedir. User2 de bu işlemleri yaparak konuşmaya başlarlar.

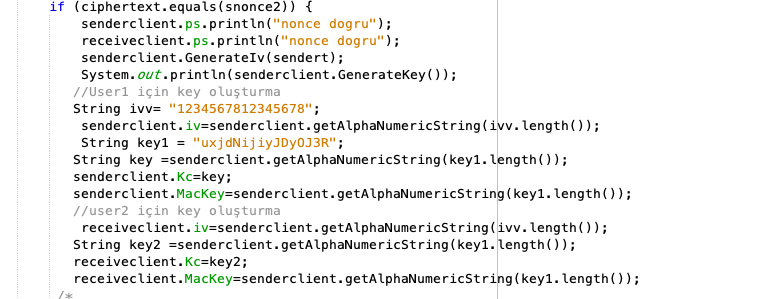
User1 ile User2 karşılıklı olarak birbirlerinin isimlerini girerek karşılıklı olarak konuşmaya başlamadan önce RSA kullanarak birbirleri ile Public Key paylaşımını yaptırdım. Daha sonra yazmış olduğum Handshake metodunun içinde paylaşılan public keyler üzerinden şifreleme işlemi yaparak nonce oluşturdum ve karşı tarafa gönderdim.



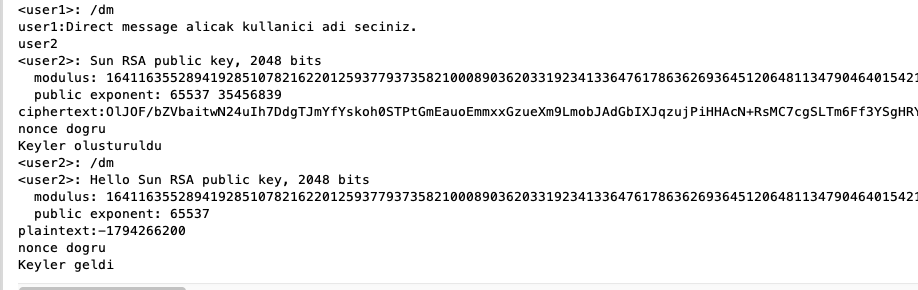
**2.Key Generation:** Gönderdiğim nonce ‘un doğruluğunu kontrol ettikten sonra Simetrik keyleri oluşturmaya başladım.



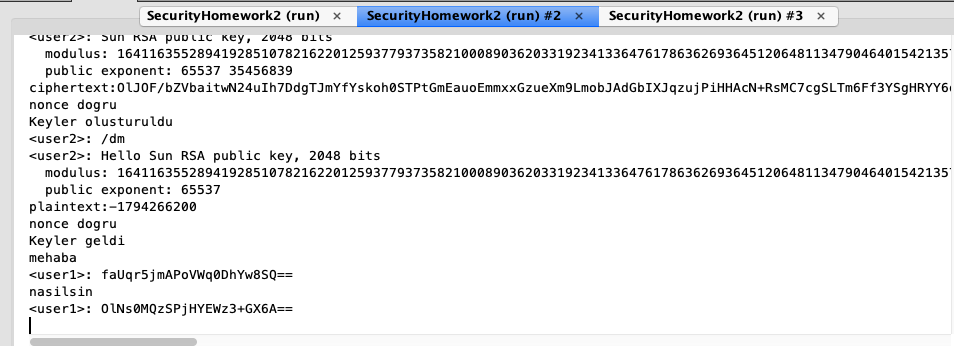
Keyleri oluştururken aldığım hatalar sonucunda AES 128 bit için İnitial vector ve Key length i elle vererek bu length size kadar random text oluşturarak istenen Keyleri ve İnitial vectorleri oluşturdum ve karşı tarafa gönderdim.



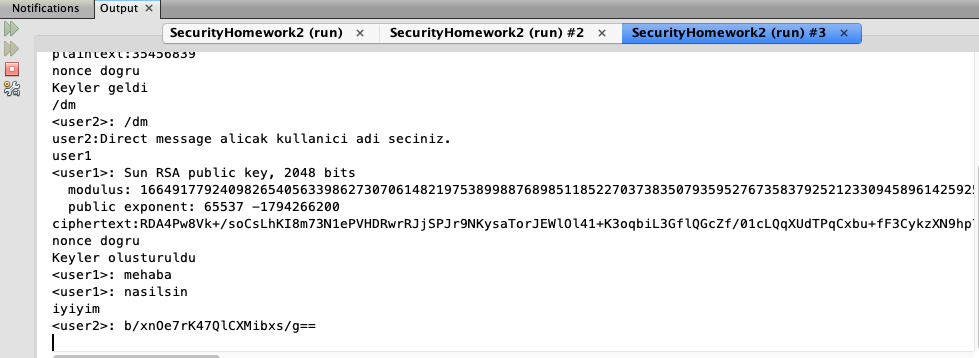
User1 ve User2 nin karşılıklı key paylaşımı ve nonce karşılaştırılması yapıldıktan sonra ki ekran çıktısı.



**3.Message Encryption:** Önceki bölümde oluşturup göndermiş olduğum Keyler ve İnitial vectorleri kullanarak AES 128 Bit (CBC) yöntemiyle birbirlerine göndermiş oldukları mesajları şifreledim.User1 in attığı mesajlar şifrelenerek karşı tarafa gönderdim ve şifreli halini yazdırdım.



User2 tarafına gelen mesajlar aşağıdaki gibidir. User1 den gelen mesajlar doğrudan şifreleri çözülerek gelmiştir ve User2’nin göndermiş olduğu mesajlar da şifreli hali yazdırılarak karşı tarafa gönderilmiştir.



**4.REFERANS**

Projede Soket Programlama kısmından yardım aldığım yer-<https://github.com/ilyasaglar/Socket-Programming-Simple-Messaging-Application>

AHMET İHSAN SAVAŞ

14121221028