Bilgi Güvenliği Ve Kriptoloji
Dersi Proje Raporu

Öğrenci Adı ve Soyadı: Fatih Üstün

Öğrenci No: 397204

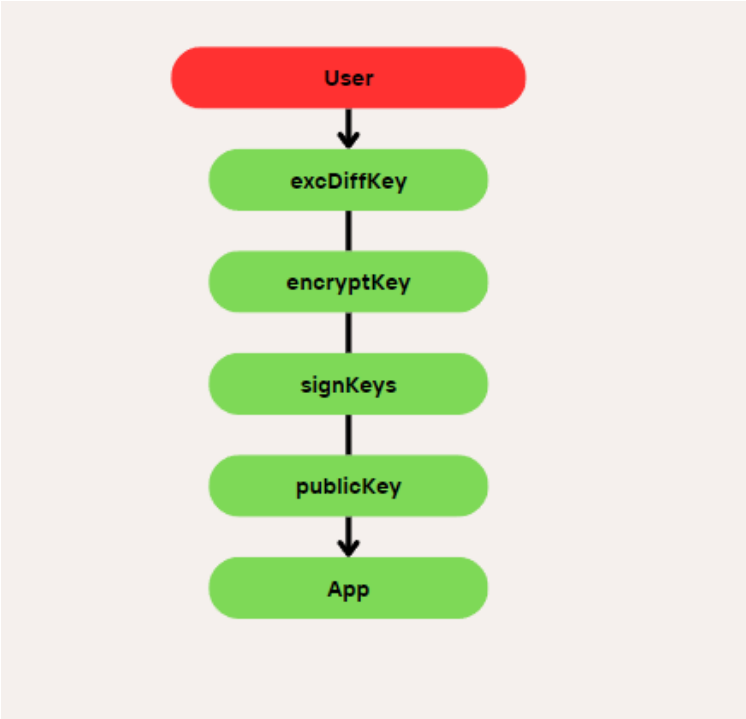
Öğrenci İletişim Bilgileri

Cep tel: 0539 250 3537

***e-mail adresi,**

| fatihustunx@gmail.com

App Of Cryptology projesi uçtan uca şifreli haberleşme uygulamasıdır. Proje 2 kişi arasında haberleşmeyi güvenli bir şekilde sağlamaktadır. Proje de açık anahtar paylaşımı için Diffie Hellman, simetrik şifreleme için Data Encryption Standart, özet için HmacSha512, imza için ise Eliptic Curve algoritması kullanılmaktadır.



User nesnesi kişileri temsil etmektedir. Bir kişi excDiffKey, encryptKey, signKeys, publicKey ve App özelliklerine sahiptir.

ExcDiffKey -> Diffie Hellman algoritmasında kullanılmak üzere kişiye özel bir numarayı temsil eder.
EncryptKey -> Des algoritmasında kullanılacak anahtarı temsil etmektedir. Paylaşım ile elde edilir.

SignKeys -> İmzalama yapmak için bir anahtar çiftini temsil eder. Private ve Public olarak 2 anahtardır.

PublicKey -> Farklı bir kullanıcının açık imza anahtarını temsil etmektedir.

App -> App ise kullanıcının ismini temsil etmektedir.

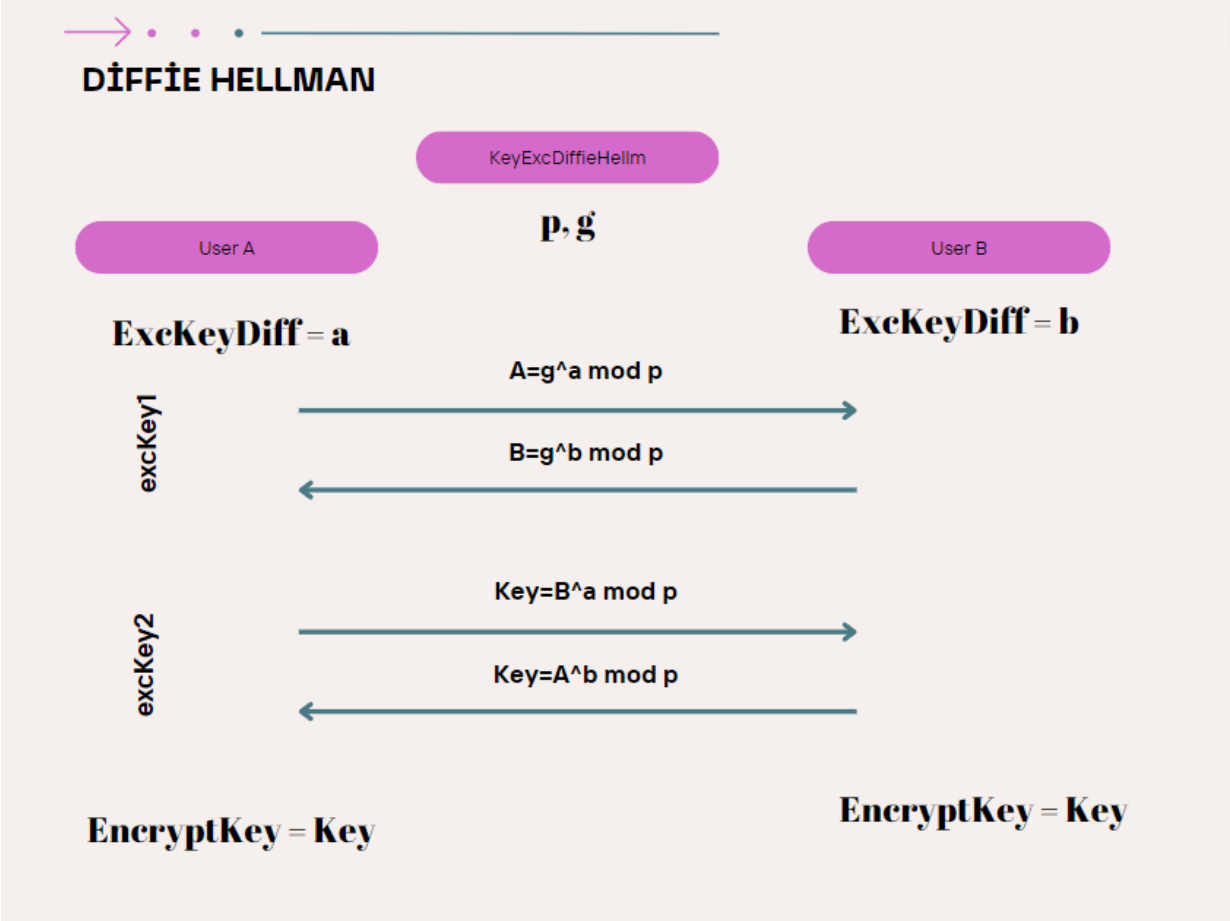
Bir kullanıcı oluşturulduğunda ExcDiffKey ve SignKeys otomatik ve rastgele olarak oluşturulmaktadır. App, yani isimleri ise kullanıcı oluşturulduktan sonra atanmaktadır. Bu özelliği kullanıcı kendisi belirler.

EncryptKey des algoritmasında kullanılacak anahtar, mesaj göndermek isteyen kullanıcı ve alıcı kullanıcı arasında Diffie Hellman açık anahtar paylaşım alogirtması ile üretilir. Bu algoritmada ise her iki kullanıcının excDiffKey'i kullanılarak EncryptKey oluşturulur. Hash için karışıklık olmaması sebebi ile farklı bir anahtar tanımlaması yapılmamıştır. Bunun için de EncryptKey kullanılmaktadır.

Bir sonraki adım da ilgili algoritmalar detaylarıyla açıklanmaktadır.

Bir kullanıcı farklı bir kullanıcı ile haberleşirken mesajlar şifrelenmektedir. Bu şifreleme Data Encryption Standart algoritması ile sağlanmaktadır. Fakat bu algoritma simetrik bir algoritmadır. Bir açık anahtar ile metin şifrelenebilir ve aynı anahtar ile deşifre edilebilir. Bu sebep ile anahtarın iki kullanıcı arasında gizli bir şekilde paylaşılması gerekmektedir. Bunun için Diffie Hellman algoritması kullanılmaktadır. Bir kişi farklı bir kişi ile mesajlaşmadan önce bu alogirtma ile anahtar paylaşımı yapmalıdır. Bu alogirtmanın projemizdeki akış şeması şu şekildedir.

Alg. Diffie Hellman.



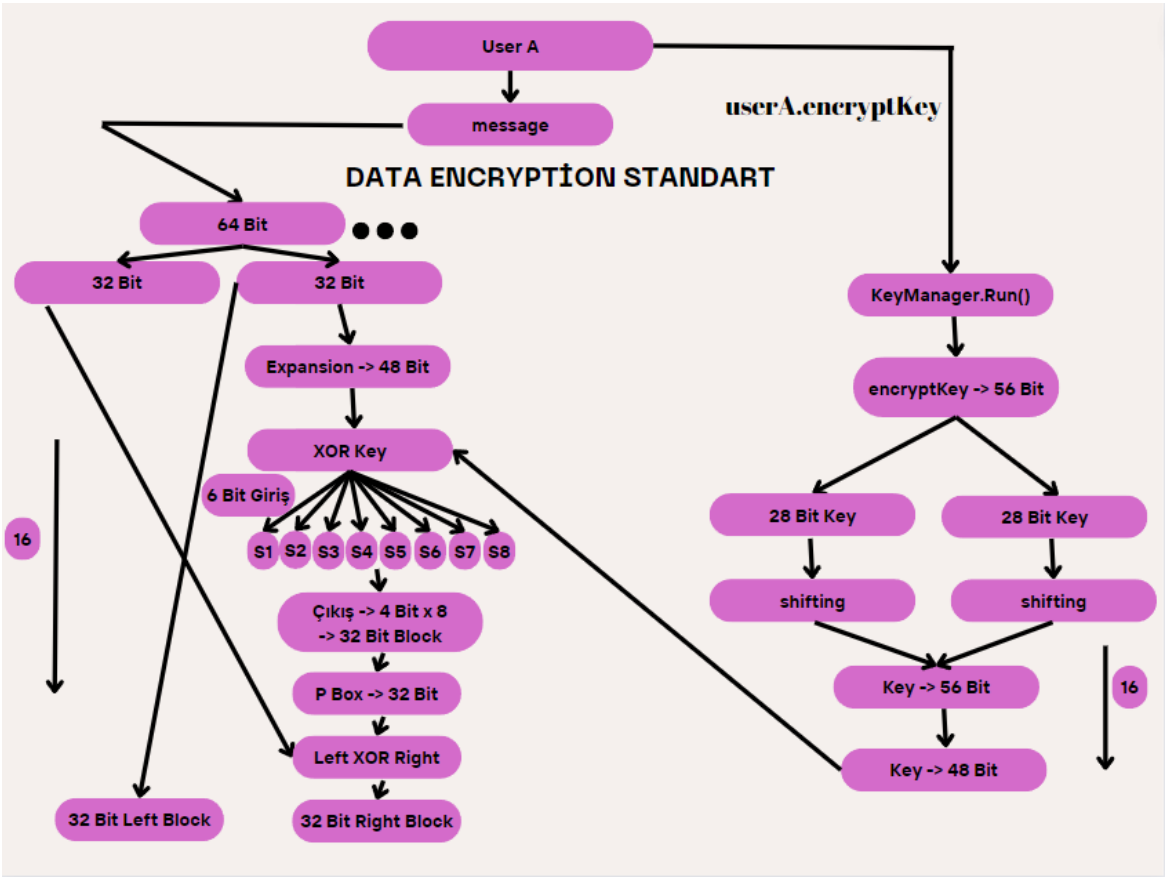
UserA ile UserB haberleşmek istediğinde öncelikle anahtar payalışımı yapılmaktadır. Bu paylaşım projede KeyExcDiffieHellm.java sınıfında yapılmaktadır. Bu sınıf p ve g olarak iki ayrı sayı içermektedir. Ve bu sayılar her kullanıcı için publictir. Ayrıca p asal sayıdır.

Öncelikle userA kendisinin sahip olduğu bir özel anahtar ile excKey1 deki işlemi yapmaktadır. Projede bu özel anahtar ExcKeyDiff olarak geçmektedir. Sonrasında userB de aynı işlemi kendi özel anahtarı ile gerçekleştirmektedir.

İkinci olarak kullanıcılar kendi özel anahtarıyla üs aldıkları ve birbirlerine gönderdikleri bu 2 sayıyı yine her kullanıcı kendi özel anahtarını kullanarak excKey2 işlemini yapar. Böylece g^a^b ve g^b^a işlemi gerçekleşmiş olur. İşlemlerin sonucundaki sayı her iki kullanıcıda bulunan Encrypt Key'e atanır. Böylece her ikisinde de aynı anahtar olur. Bu anahtar 64 bit olup projenin bir sonraki aşaması olan DES algoritmasında kullanılmak üzere her iki kullanıcıda da bulunmaktadır.

Artık kullanıcıların her ikisinde de Des için kullanılacak olan EncryptKey bulunmaktadır. UserA bir mesaj göndermek istediğinde bu key ile mesajını şifreler. Des adımları aşağıdaki şemada gösterilmiştir.

Alg. Data Encryption Standart.



İlk olarak userA anahtarı ile Des algoritmasında kullanılması için anahtarlar üretmektedir. Bunun için projede KeyManager.java sınıfı kullanılmaktadır. 64 Bitlik anahtar alınır her 8. Bit yok edilerek 56 Bite indirgenir. 56 Bitlik anahtar ortadan ikiye ayrılır ve sağ sol olarak 2 anahtar elde edilir. Bu iki anahtar Des alogirtması için tanımlı olan kaydırma özelliklerine ve sayısına göre 1'er veya 2'şer bit olarak kaydırılır. Sonrasında birleştirilir ve 48 bite indirgenir. Bu işlemler kaydırma aşmasından sonrası için 16 kere yapılır. Bu döngü sonrasında Des de kullanılmak üzere 16 adet anahtar elde edilmiş olunur.

İkinci olarak Des alogiritması uygulanır. Bu algoritma projede EncryptDesManager.java sınıfındadır. Mesaj alınır ve 64 bitlik bloklara bölünür. Her blok için soldaki işlemler 16 kere uygulanır. En sonunda ise birleştirilerek şifreli mesaj oluşturulur. İşlem adımları şu şekildedir.

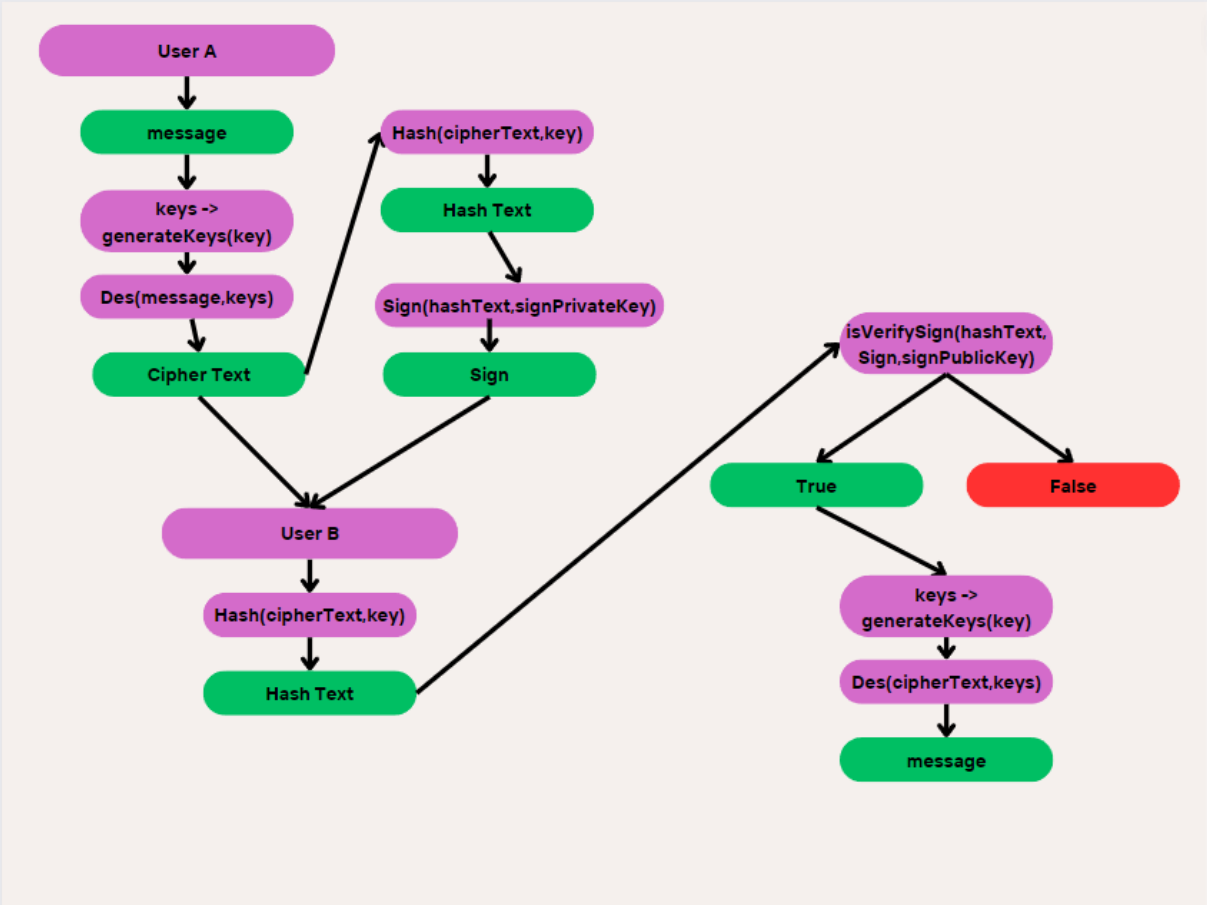
64 Bitlik blok 32 bit olarak sol ve sağ olarak ayrılır. Sağ blok genişletme permütasyonu ile 48'e genişletilir. 48 Bitlik blok üretilen anahtar ile her döngüde farklı bir anahtar olmak üzere xor işlemine sokulur. Sonrasında 48 bitlik işlem sonucu 6 bitlik girişler olmak üzere 8 farklı S kutusu ile işleme sokulur. Her kutudan 4 bitlik bir çıkış toplamda 32 bitlik bir blok elde edilmiş olunur. Bu blok P Box ile permütasyona sokulur. Daha sonra sol blok ile xor'lanır. Yeni sağ blok bu işemin sonucu olur. Yeni sol blok ise sağ blokun bu işlemlere girmeden önceki ilk halidir.

Data Encryption Standart algoritmasında bu işlemler 16 defa tekrarlanır. En son döngüde sağ ve sol blok yer değiştirilir ve toplanır. Mesajın her 64 bitlik bloğu için bu işemler gerçekleştirilir. Oluşan bloklar en sonunda tekrardan toplanır ve şifreli mesaj elde edilmiş olunur.

UserA şifreli mesajı elde ettikten sonra bu mesajı hash fonksiyonuna gönderir. Projede key karmaşıklığı olmaması için hash içinde des de kullanılan key kullanılmaktadır. Hash fonksiyonunda özet bilgi elde edildikten sora bu özet bilgi userA'nın private sign key'i kullanılarak imzalanır.

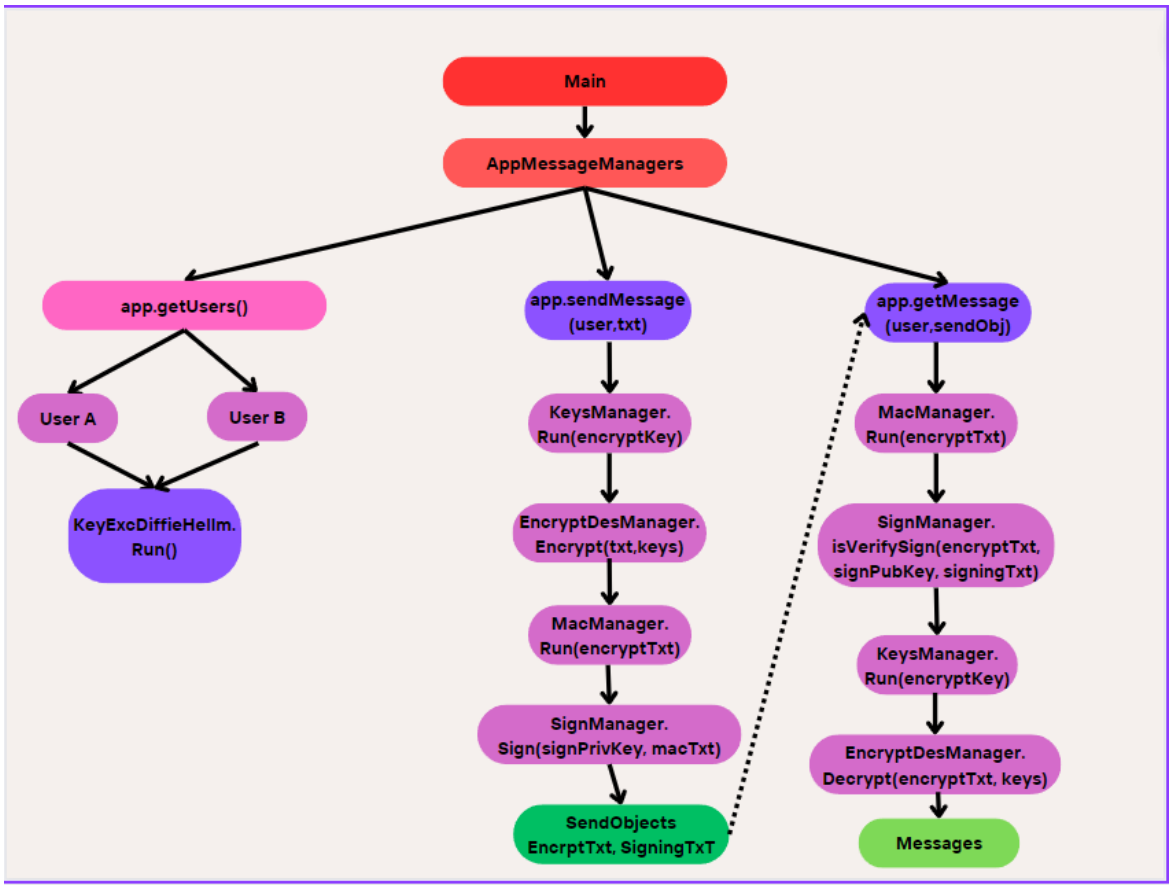
Son olarak ise UserA şifreli mesajı ve imzayı UserB'ye gönderir.

Akış.



UserB UserA'dan gelen şifreli mesajı ve imzayı alır. Öncelikle mesaj bütünlüğü için şifreli metni hashler. Sonrasında bu özet bilgiyi, userA'nın imzasını ve userA'nın public imza Key'ini kullanarak imza doğru mu değil mi diye kontrol eder. Eğer doğru ise Encrypt Key'i kullanarak Des alogirtması için anahtarlar oluşturur. Bu anahtarları ve şifreli metni Des'e gönderir. Des de decrypt için anahtarları ters olarak vermesi gerekmektedir. Bu işlem sonucunda ise şifreli mesajı decrypt eder ve mesaja ulaşır.

Akış 2.



App Of Cryptology de ki yapıların daha iyi anlaşılması için Akış 2 oluşturulmuştur.

```
        <-- Encrypt Keys & Signing Public Keys Exc... -->

Ali -> hi

Encrypt Txt : ½ÐoKµù¤O
Signing Txt : [B@4ca8195f

Get Message --> hi

Veli -> how are u ?

Encrypt Txt : öP¼iZ¦~Áû¿ù8Ä
Signing Txt : [B@490d6c15

Get Message --> how are u ?

Ali ->
```

Son olarak proje de Ali ve Veli kullanıcıları oluşturulmıştur. Bu kullanıcılar oluşturulduktan sonra aralarında aralarında anahtar paylaşımı yapılmıştır. Mesajlar şifrelenip imzalanmış ve ilgili anahtarlar kullanılarak deşifre edilmiştir.Böylelikle uçtan uca güvenli ve gizli bir şekilde haberleşme sağlanmıştır.