T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

MOBİL ORTAMLAR İÇİN GÜVENLİ VERİ İLETİŞİM KANALI GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

HÜSEYİN BODUR

HAZİRAN 2015

DÜZCE

KABUL VE ONAY BELGESİ

Hüseyin BODUR tarafından hazırlanan MOBİL ORTAMLAR İÇİN GÜVENLİ VERİ İLETİŞİM KANALI GELİŞTİRİLMESİ isimli lisansüstü tez çalışması, Düzce Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ..... sayılı kararı ile oluşturulan jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans olarak kabul edilmiştir.

Üye
(Tez Danışmanı)
Doç Dr. Resul KARA
Düzce Üniversitesi

Üye
Unvan, Ad Soyad
Üniversitesi

Üye
Unvan, Ad Soyad
Üniversitesi

Üye
Unvan, Ad Soyad
Üniversitesi

Üye
Unvanı, Ad Soyad
Üniversitesi

Tezin Savunulduğu Tarih : .....

ONAY

Bu tez ile Düzce Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu Hüseyin BODUR'un Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans derecesini almasını onamıştır.

Prof. Dr. Haldun MÜDERRİSOĞLU
Fen Bilimleri Enstitüsü Müdürü

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

09 Haziran 2015

İmza

Hüseyin BODUR

Sevgili Aileme

TEŞEKKÜR

Yüksek lisans öğrenimim ve bu tezin hazırlanmasında süresince gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Doç. Dr. Resul KARA'ya en içten dileklerimle teşekkür ederim.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili aileme ve çalışma arkadaşlarıma sonsuz teşekkürlerimi sunarım.

Bu tez çalışması, Düzce Üniversitesi BAP-2015.06.01.302 numaralı Bilimsel Araştırma Projesiyle desteklenmiştir.

09 Haziran 2015 **Hüseyin BODUR**

İÇİNDEKİLER **Sayfa**

ŞEKİL LİSTESİ

ŞEKİL LİSTESİ (devam)

ÇİZELGE LİSTESİ

SİMGELER VE KISALTMALAR

AES	Advanced Encryption Standard (Gelişmiş Şifreleme Standardı)
EC	Elliptic Curve (Eliptik Eğri)
ECC	Elliptic Curve Cryptology (Eliptik Eğri Şifreleme Sistemi)
ECDH	Elliptic Curve Diffie Hellman (Eliptik Eğri Diffie Hellman)
ECDLP	Elliptic Curve Discrete Logarithm Problem (Eliptik Eğri Ayrık Logaritma Problemi)
DES	Data Encryption Standard (Veri Şifreleme Standardı)
GCM	Google Cloud Messaging (Google Bulut Mesajlaşma)
MS	Milisaniye
NITS	National Institute of Standards and Technology (Ulusal Standartlar Enstitüsü)

ÖZET

MOBİL ORTAMLAR İÇİN GÜVENLİ VERİ İLETİŞİM KANALI GELİŞTİRİLMESİ

Hüseyin BODUR
Düzce Üniversitesi
Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi
Danışman: Doç. Dr. Resul KARA
Haziran 2015, 76 sayfa

Kriptoloji bilgi güvenliğini içeren bilim dalıdır. Bir verinin iletim ortamına gönderilmeden önce okunmaz hale getirilmesi, hedefine ulaştıktan sonra ise karşı tarafın veriyi yeniden okunabilir haline çevirebilmesi için kullanılan şifreleme konularının genel adıdır. Veri güvenliğinin üst düzey olduğu alanlarda veri şifreleme algoritmaları ön plandadır.

Bu çalışmada web servisleri aracılığıyla mobil platform üzerinde güvenli haberleşmenin sağlanabilmesi için, iletim ortamına bırakılacak ve ortamdan alınacak verilerin şifreleme algoritmalarıyla şifrelenip aktarılması sağlanmış, iki asimetrik şifreleme sistemi olan RSA [1] ve Eliptik Eğri [2] şifreleme sistemleri ile iletim verisi üzerinde şifreleme ve şifre çözme işlemleri uygulanıp, kıyaslamaları yapılmıştır. Bu yapıların performans karşılaştırmaları incelenip, alınan sonuçlar analiz edilmiştir.

Analizlerden elde edilen bilgilere göre, mobil cihazlarda eliptik eğri şifreleme sisteminin RSA'ya oranla daha küçük anahtar boyutu ile daha güvenli, daha karmaşık şifreleme yaptığı belirlenmiştir. Anahtar üretme, şifreleme, şifre çözme, şifreli mesajı iletme sürelerinin karşılaştırılması sonucunda daha küçük anahtar boyutlarına sahip Eliptik eğri siteminin daha kısa sürede sonuca gittiği görülmüştür. Doğru algoritma ve anahtar boyutlarının seçiminde, yapıların birbirlerine göre avantaj ve dezavantajları ifade edilmiştir.

Anahtar sözcükler: Android, Eliptik Eğri, Mesajlaşma, RSA

ABSTRACT

DEVELOPMENT OF SECURE DATA COMMUNICATION CHANNEL FOR MOBILE ENVIRONMENTS

Hüseyin BODUR
Duzce University
Graduate School of Natural and Applied Sciences, Departmant of Computer Engineering
Master of Science Thesis
Supervisor: Assoc. Prof. Dr. Resul KARA
June 2015, 76 pages

Cryptology is the branch of science that contains the information security. It is the general name of the encryption issues used for making the data unreadable before sending it to the transmission medium and translating it into readable form again after reaching the other side. Data security algorithms are in the foreground in the areas which have high level of data security. In this study, transfering and encryption of datas which were leaved and received from the transmission medium by encryption algorithms were ensured in order to provide secure communications on mobile platforms through web services.

Encryption and decryption operations were applied on transmitted data with RSA[1] and Elliptic Curve Cryptography [2] which are two asymmetric encryption systems and the comparisons were made.

The information obtained by the analysis, it was expressed that elliptic curve cryptosystem provides safer and more sophisticated encryption with smaller key sizes compared to the RSA cryptosystem on mobile platforms. A result of the comparison of key generation, encryption, decryption, encrypted message transmission times, it was stated that encryption, decryption and transmission times with Elliptic curve system which has smaller key sizes lasted shorter than RSA system. In the selection of the right algorithm and key sizes, it was stated that what the advantages and disadvantages of structures were according to each other.

Keywords: Android, Eliptic Curve, Messaging, RSA

EXTENDED ABSTRACT

DEVELOPMENT OF SECURE DATA COMMUNICATION CHANNEL FOR MOBILE ENVIRONMENTS

Hüseyin BODUR
Duzce University
Graduate School of Natural and Applied Sciences, Departmant of Computer Engineering
Master of Science Thesis
Supervisor: Assoc. Prof. Dr. Resul KARA
June 2015, 76 pages

1. INTRODUCTION:

Cryptology is the branch of science that contains the information security. It is the general name of the encryption issues used for making the data unreadable before sending it to the transmission medium and translating it into readable form again after reaching the other side. Data security algorithms are in the foreground in the areas which have high level of data security. Essentially, the usage areas of RSA [1] and Elliptic Curve system [2] were investigated with a comprehensive literature search in the study. Implementation and comparison of these two algorithms on mobile platform were determined as the purpose. Then, the history of cryptography and the meanings of some concepts which are often used in the cryptography were mentioned. In this study, transfering and encryption of datas which were leaved and received from the transmission medium by encryption algorithms were ensured in order to provide secure communications on mobile platforms through web services.

Encryption and decryption operations were applied on transmitted data with RSA and Elliptic Curve Cryptography which are two asymmetric encryption systems and the comparisons were made.

The performance comparisons of these systems were examined and the results were analyzed.

2. MATERIAL AND METHODS:

Before giving the technical details of the study, the algorithms used in the encryption field and the working structures were discussed. Then, the information about Google Push Notification technology which is used in the application was given and an example about this subject was performed. Google Cloud Messaging is a technology that allows real-time communication. It is not necessary that the device used goes to server at regular intervals and updates its information thanks to this technology. The device is triggered by the server when an event occurs. The source codes of the examples performed by this technology were shared in media environment. The application was explained in a detailed way by using the RSA algorithm. How the encryption process is realized with RSA and Elliptic Curve system was mentioned. On the server side, how the data is sent to the server and stored in the server was explained.

3. RESULTS AND DISCUSSIONS:

On the sender side, the message which is transferred to another mobile device via server on the mobile device is subjected to encryption processing with either Elliptic Curve system or RSA system. When the stored data on the server is transmitted to the receiver side, the application executes the decryption process automatically to obtain the clear text. Secret and public keys which are two public key infrastructure are used at this point. Sender side encrypts the text with the public key of receiver side. The receiver side decrypt the encrypted data with its own private key. At this point, both RSA and Elliptic Curve systems' average key creation time, their encryption time and their transmission time of encrypted data are calculated. These calculated times are used in the algorithms comparison process.

4. CONCLUSION AND OUTLOOK:

The processor structures and the power capabilities of mobile devices is different and limited. Therefore, key generation time, encryption time, decryption time and transmission time should be as short as possible. The shorter processing time is, the lower complexity of the system is. This situation affects the device's processing capacity and battery consumption directly. Reduced processing times means that mobile devices makes less processing. In this case, both the processor of the mobile device gets less tired and the battery consumption is reduced. Security issue must not be forgotten at this point. The user should not concede security when s/he prefers a system which has less

complexity. Encrypted data must be resistant against the attacks. When RSA and Elliptic Curve systems are compared, it was seen that Elliptic Curve system provides the same security level with lower key values than RSA. Namely, it means that Elliptic Curve system provides the same level of security with less complexity. When we compare two encryption structures with regards to their times of key creation, encryption and transmission, the times calculated with elliptic curve algorithm are seen to be shorter than the times calculated with RSA. This case shows that Elliptic Curve system is one step ahead when it is compared with the RSA system. However, both systems have advantages and disadvantages according to where they are used.

1. GİRİŞ

1.1. AMAÇ VE KAPSAM

İnternet üzerinden gönderilen veri paketlerinin, herkesin erişimine açık olan networklerden geçmesi, bu paketlere saldırı düzenlenmesini ve içeriklerine erişilmesini mümkün kılar.

Özellikle içerisindeki bilgilerin son derece önemli olduğu paketlerin transferi büyük bir kaygıya neden olur. Bu paketlerin güvenliği sağlanmadan internet üzerine aktarılması, yazışmalarda, ticaret işlemlerinde vb. birçok işlemde kullanılması güvenli değildir. Bu noktada "Bilgi Güvenliği" kavramları devreye girmektedir. Bilgi güvenliği gizlilik, kimlik denetimi (asıllama), bütünlük, inkâr edememe gibi matematiksel yöntemlere sahiptir.

Bu yöntemler kriptografinin önemli konularıdır ve bilginin aktarımı esnasında karşılaşılabilecek aktif ya da pasif saldırılardan bilgiyi koruma amacı taşırlar.

Bu yöntemler aşağıda açıklanmıştır [3];

- *Gizlilik:* Bu servis iletilen verinin sadece yetkisi olan kullanıcı tarafından erişilebilir olmasıdır. Veri diğer tüm ortam için özel ve gizlidir.

- *Bütünlük Sağlama:* İletilen veri sadece yetkili kişiler tarafından değiştirilir, bunun dışında veri bütünlüğü korunur. Bütünlük koruması aktif saldırılar ile ilgili bir özelliktir. Bu nedenle bütünlüğün bozulduğunu tespit etmek, bütünlüğü sağlamaktan daha önemlidir.

- *Kimlik Denetimi (Asıllama)* : Bu özellik veri kaynağının doğruluğunu kontrol eder. Güvensiz ortamdan gönderilen verinin kaynağı, içeriği, gönderildiği saati, gönderen kaynağın saati gibi parametreler asıllanır. Özellik iki ana alt dala ayrılır. Bunlar, bütünlük asıllaması ve veri kaynağı asıllamasıdır.

- *İnkâr Etmeme:* Bu özellik haberleşen noktaların daha önce gönderdikleri verileri ve yaptıkları istekleri inkâr edememelerini sağlar.

Şifreleme yöntemleri eskiden hükümet çalışmalarında yâda askeri çalışmalarda

kullanılırdı; ancak günümüzde gizlilik ve şahsi bilgilerin korunması noktalarında vazgeçilmez bir konuma sahip hale gelmiştir. Özellikle e-ticaret, finans, bankacılık, iletişim ve kamu hizmetleri gibi birçok alanda şifreleme yöntemleri verilerin iletimi ve alımı noktasında aktif rol oynar. İnternet teknolojilerinde kullanılan teknoloji yöntemlerinin sürekli gelişmesi, bu yöntemlerin uygulanması esnasında gerekli güvenlik tedbirlerinin artmasını da beraberinde getirir. Bu ihtiyaçlar neticesinde kriptoloji alanında yapılan çalışmalarda da artış meydana gelir.

Kriptoloji kısaca şifre bilimidir. Amacı güvensiz ortam içerisinde, güvenli bir yol oluşturarak haberleşmek isteyen noktaların iletişim güvenliğini sağlamaktır [4]. Kriptografi ve kriptoanalizin bir arada olduğu bilim dalıdır. Kriptanaliz, kriptografik sistemlerin kurduğu yapıları inceyip, bu yapıları çözmeye çalışır. Kriptografi ise güvenilirlik, veri bütünlüğü, gizlilik, kimlik denetimi, geçerlilik gibi matematiksel yöntemlerin bir araya geldiği çalışma alanıdır.

Bu çalışmada mobil ortamda uçtan uca veri güvenliğini sağlayacak şifreleme yapıları geliştirerek, bu yapılar yardımıyla güvenli iletişim yoluna sahip, yaşanabilecek güçlü saldırılara karşı veri bütünlüğünü ve tutarlılığını koruyan, sızmalara karşı duyarlı, bir haberleşme sisteminin oluşturulması hedeflenmiştir. Bu amaca ulaşma noktasında iki açık anahtarlı şifreleme algoritmasının hız, şifreleme, şifre çözme ve anahtar oluşturma zamanları açısından analizlerini gerçekleştirdik.

Piyasada bu amaca yönelik olan ve internet üzerinden karşılıklı mesajlaşma altyapısını kullanan sistemler incelendiğinde, bu işlevselliğe sahip Whatsapp, Facebook Messenger gibi uygulamaların mevcut olduğu görülür. Uygulamaları yayınlayan şirketlerin söylemlerine göre bu uygulamalar göndericinin gönderdiği metni iletim ortamına bırakmadan önce belirli şifreleme algoritmalarından geçirirler. Bu uygulamalarda ortama bırakılan metin anlamsız veriler bütünü şeklindedir. Bu noktada iletim ortamından alıcı tarafa geçen veri öncelikle deşifreleme işlemine tabi tutulur ve alıcı tarafa metnin anlamlı formu iletilir. Bu tarz uygulamalar oldukça gelişmiş olmalarına rağmen, resmi olarak uygulamaların kod yapılarına kullanıcıların veya üçüncü şahısların girebilmesi ve kullandıkları teknolojileri inceleyebilmeleri mümkün değildir. Söylenilen şifreleme alt yapısının ne ölçüde güvenli olduğu ise bir başka tartışma konusudur. Bu noktada mobil ortamda hedefimiz bir şifreleme yapısı oluşturup, yapı içerisinde günümüzde güvenilir kabul edilen şifreleme algoritmalarını karşılaştırarak literatüre ve

güvenli haberleşme alanına bir katkı sağlamak olmuştur.

Bu tez çalışması kapsamında bir mesajlaşma uygulaması geliştirilmiştir. Bu uygulama ile öncelikle iki kişinin internet ortamında gerçek zamanlı mesajlaşması sağlandı. Bunun için Google Cloud Messaging teknolojisi kullanıldı. Anlık iletilen mesajlar hem mobil cihazın veritabanında hem de internet üzerindeki veritabanında tutuldu. Kullanıcıdan tek seferliğe mahsus uygulamaya giriş yapacakları kullanıcı adı ve şifre bilgileri istendi ve bu bilgilerle uygulama güvenliğinin bir seviye daha artırılması hedeflendi.

Gerçek zamanlı karşılıklı iletişim sağlanıp, güvenli mesajlaşmayı gerçekleştirmek amacıyla RSA ve Eliptik Eğri şifreleme yapıları kullanılarak karşılıklı iletişimde aktarılan mesajlar şifrelendi. Ardından bu iki sistemin birbirlerine göre avantaj ve dezavantajlarının neler olduğuna, hangi durumlarda hangi şifreleme yapısının kullanılması gerektiğine değinildi.

1.2. LİTERATÜR TARAMASI

RSA ve Eliptik eğri şifreleme sistemleri verilerin iletiminde diğer şifreleme sistemlerine oranla daha güvenli oldukları ve henüz güvensizlikleri ispatlanmadığı için birçok farklı alanda kullanılmış ve kullanılmaya da devam etmektedir. Kullanılacak alanlarının yazılımsal ve donanımsal performansları bu yapıların içlerinde kullanılacak şifreleme yapısını doğrudan belirleyen önemli kriterlerdir.

Bu kriterler doğrultusunda Islam ve Biswas tarafından yapılan çalışmada, ağlar üzerindeki hareketli cihazlar ile ağın karşılıklı kimlik doğrulaması üzerine bir çalışma yapılmış, yapılan bu çalışma ile ağın güvenlik açıkları belirlenmiş ve bu açıkların Eliptik eğri şifreleme algoritması kullanılarak nasıl yok edildiğine değinilmiştir. Bu sayede güvenli bir iletişimin nasıl oluşturulacağını inceleyip, algoritmanın düşük güç tüketimi yaparken güçlü güvenlik seviyelerine sahip olduğunu belirtmişlerdir. Algoritmanın ağ üzerinde yapılabilecek internet bankacılığı, online alışveriş gibi önemli işlemler için oldukça güvenilir ve uygun olduğunun altını çizmişlerdir [33].

Bir başka çalışmada Abi-Char ve arkadaşları Eliptik eğri ayrık logaritma problemine dayanan bir protokol geliştirerek kablosuz iletişimde maruz kalınan saldırılardan korunan güvenli bir iletişim protokolü oluşturmuşlardır. Protokolde Eliptik eğri

algoritmasıyla birlikte ElGamal imza algoritmasından yararlanılarak sistemin direncini arttırmışlardır. Sistemin sunucu atakları ve ortadaki adam atakları gibi ataklara karşı olan direncini gösterip, geliştirilmiş olan diğer protokollerle (B- Simple Password Exponential Key Exchange, Secure Remote Password, Elliptic Curve - Secure Remote Password vb.) karşılaştırılmasını yapmışlardır [34].

Sahana ve Misra yaptıkları çalışmada kablosuz algılayıcı ağlar üzerinde güvenlik ve enerji verimliliği için asimetrik protokol olan RSA algoritması üzerinde optimize hesaplama yaparak enerji verimliliği sağladıklarını ve RSA algoritmasının enerji ihtiyacını simetrik protokollerin ihtiyacına benzer seviyelere indirdiklerini belirtmişlerdir [47]. Bir başka çalışmada, kablosuz ağda ortadaki adam saldırısından korunmak için hesaplama karmaşıklığı fazla olan RSA algoritması yerine aynı görevi daha düşük karmaşıklıkla sağlayan Eliptik Eğri algoritması kullanılmıştır [48].

Ravikumar ve Udhayakumar Eliptik Eğri algoritmasını çoklu elektronik işlemlerinde kullanmışlardır. Güvenli bir banka uygulaması içeren küçük bir örnek yapmışlardır. Çalışmalarında, diğer açık anahtarlı sistemlerle karşılaştırıldığında aynı güvenlik seviyelerini küçük anahtar boyutlarıyla sağladığından Eliptik Eğri sisteminin ümit verici olduğunu belirtmişlerdir. Eliptik Eğri algoritmasının diğer kriptolojik algoritmalara göre çok daha güvenli olduğuna, anahtar boyutunun küçük olmasından dolayı daha az anahtar depolaması ve yüksek hız sağladığına değinmişlerdir [35].

Saini ve Vaisla dijital imzalama şifrelemenin birleşimi olan Signcryption üzerine bir çalışma yapmışlardır. Signcryption'ın metinler üzerine yapılmasına ilave olarak resimler üzerine de uygulanmasını sağlamışlardır. Çalışmalarında, alıcı taraf resim üzerinde signcryption işlemi yapıp karşı tarafa göndermiş, gönderici taraf ise unsigncrypt işlemi yaparak orjinal resmi yeniden elde etmiştir. Signcryption işlemi için Eliptik Eğri algoritması kullanılmıştır. Eliptik Eğri algoritmasının küçük anahtar boyutu ile yüksek güvenlik sağladığı, bu sayede hesaplama ve iletişim maliyetinin azaldığı vurgulanmıştır. Örnek uygulamada resim öncelikle binary formata çevrilip resmin boyutu küçültülmüş ardından signcryption işlemi uygulanmıştır. Çalışmada resim göndermenin ve resmi tekrar orijinal hale getirmenin kolay olduğu, askeri ve medikal amaçlar için kullanılabileceği vurgulanmıştır [36]. Benzer çalışmada Zhao ve arkadaşları RSA algoritmasını kullanarak dijital resim üzerinde şifreleme ve şifre çözme işlemi yapmışlar ve bu resmin kablosuz ağ ortamında algılayıcı tabanlı bilgi gizleme

modunda aktarımından bahsetmişlerdir [49]. Bir başka çalışmada Gupta ve arkadaşları Eliptik Eğri kriptolojisini kullanarak resim şifreleme üzerine bir çalışma yapmışlar. Çalışmada her pikseli, Eliptik Eğri noktasına, her noktayı ise şifreli resim pikseline dönüştürmüşlerdir [50].

Bakhtiari ve arkadaşları Eliptik Eğri kriptolojisini kullanarak yeni bir mobil ödeme sistemi tasarlamışlardır. Tasarladıkları şemada 160 bitlik ECC anahtarı kullanmış ve bu şemanın altyapısı ve detayları hakkında bilgi vermişler. Sistemde Eliptik Eğri kullanmanın sorun ve kısıtlamalarının neler olduğunu tartışmışlar [37].

Losinek ve Drahansky SMS iletim güvenliğinde simetrik ve asimetrik metotların kulanım farklılıklarını karşılaştırıp, asimetrik bir metot olan RSA şifreleme algoritması ile örnek bir uygulama geliştirmişlerdir. Örneklerinde anahtar boyutu olarak 1024 bitlik anahtar kullanmış, 1024 bitlik anahtar değerinin 128 karakterlik bir veri şifrelemeye izin vereceğini belirtmişler ve bu durumun avantaj ve dezavantajlarına değinmişlerdir [38]. Benzer çalışmada Agori ve Seral, RSA, ELGamal ve Eliptik Eğri algoritmalarını kullarak SMS şifreleme yapmış ve bu algoritmaları karşılaştırmıştır [43]. Saxena ve Chaudhari, ECDSA, RSA ve DSA algoritmaları ile SMS üzerinde dijital imza kullanarak güvenli mesajlaşma üzerine çalışmışlardır [44].

Somani ve arkadaşları RSA algoritması kullanarak bulut depolama üzerinde veri güvenliği üzerine araştırma yapmışlardır. Çalışmada gönderici taraftan alıcı tarafa verinin bozulmadan ve iletim esnasında saldırılara uğramadan geldiğinden emin olmak için RSA şifreleme algoritmasından yararlanılmıştır. Öncelikle bir hash algoritmasıyla aktarılacak verinin özeti çıkarılmış ardından çıkarılan bu özet göndericinin gizli anahtarıyla ve alıcının açık anahtarıyla şifrelenmiştir. Veri aktarımının ardından alıcı taraf önce kendi gizli anahtarını ardından göndericinin açık anahtarını kullanarak deşifreleme işlemini gerçekleştirmiş ve mesaj özetinin mesajdan elde edilen özetle aynı olup olmadığını kontrol etmiştir. Bu sayede bulut teknolojisini kullanarak gönderici ve alıcı arasında giden verinin yüksek güvenlikle korunması sağlanmış, verinin iletim işlemi sırasında zarar görüp görmediği, saldırılara uğrayıp uğramadığı konusunda bilgi sahibi olunmuştur [39]. Bulut servisleri üzerinde Rathanam ve Sumalatha dinamik ve güvenli bir depolama sistemi tasarlamışlardır. Sistemin güvenlik kısmında RSA şifreleme algoritmasını kullanmışlar, verinin güvenlikten ödün verilmeden azaltılmış hesaplama maliyeti, yer ve zaman tüketimi ile depolanmasını amaç edinmişlerdir [52].

Bir diğer çalışmada mobil cihazlar kullanarak elektronik oylama üzerine bir çalışma yapılmıştır. Alrodhan ve arkadaşları elektronik oylamanın gerçek hayatta yer bulamadığına değinmişler ve hayata geçirilmesi için sistemin ne tür özellikleri olması gerektiği vurgulamışlardır.

Ardından elektronik oylama için yapılması gerekenlerin ne olduğuna değinip, sistemi Doğrulama merkezi ve Oy verme merkezi olarak ikiye ayırmışlardır. Çalışmanın doğrulama aşamasında Doğrulama merkezine gidilip, kullanıcıdan ID bilgisi ve parmak izi örnekleri toplanmıştır. Burada kişinin gerçek kişi olup olmadığı yani sisteme kayıtlı olup olmadığının doğrulaması yapılmıştır. (Oy verecek tüm vatandaşların bu sisteme önceden kayıtlı olması gerekmektedir.)

Bu aşamanın ardından kullanıcıya ya olumlu mesaj gönderilip oy verme aşamasına geçilmiş yâda hata mesajı görüntületilmiştir.

[40,45,46]'da güvenli elektronik oylama ile ilgili çalışmalar yer almaktadır.

Melgar ve Santander QR kod, RSA şifreleme sistemi ve dijital imzalamanın birlikte kullanıldığı bir ürün dağıtım sistemi üzerinde çalışmışlardır.

Gani ve Abdurohman RSA şifreleme algoritmasını kullanarak video verisi üzerinde şifreleme işlemi yapmışlar, video formatı olarak MPEG kullanmışlardır. Videonun iletiminde güvenliğin sağlanabilmesi için şifreleme işleminin oldukça yararlı olduğuna, bu konuda esnek olarak herhangi bir şifreleme algoritmasının kullanılabilineceğine değinmişler, kaba kuvvet ataklarına karşı dayanıklı olduğundan RSA algoritmasını tercih etmişlerdir [42]. Wang ve arkadaşları ses sinyali üzerine dijital doğrulama sistemi uygulamışlar ve sinyalin sahibini doğrulayacak imzayı ses sinyalinin içerisine gömmüşlerdir. Bu sayede ses sinyali alıcı tarafa ulaştığında alıcı, sinyalin saldırıya uğrayıp uğramadığını fark edebilecek ve sinyal sahibinin kim olduğu konusunda bilgi sahibi olabilecektir. İmzalama işlemi RSA sistemi kullanılarak yapılmış ve sinyal üzerine eklenen imza orijinal ses sinyali kalitesinde bozulmalara neden olmamıştır [51].

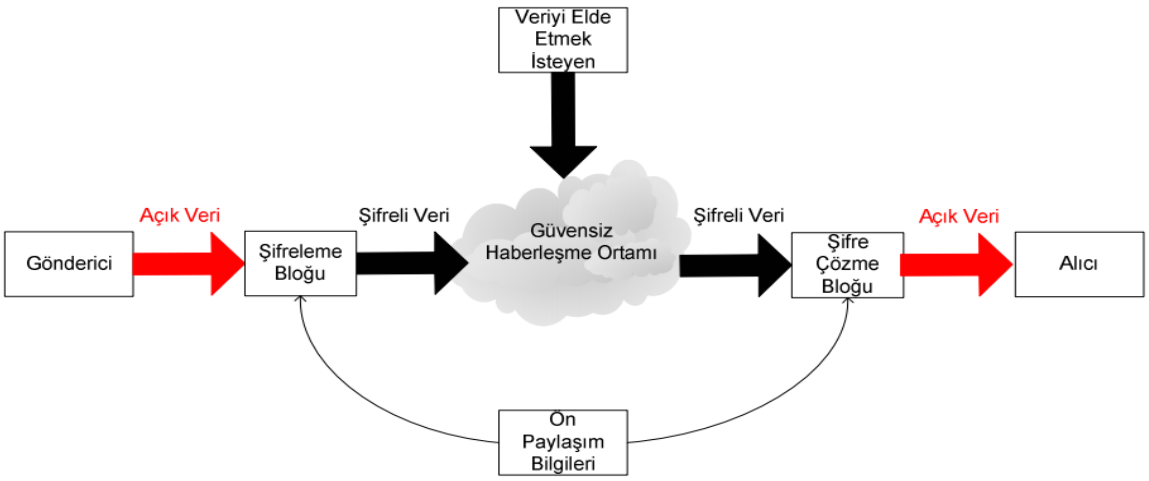
Literatürde yer alan çalışmalardan da anlaşılacağı gibi RSA ve Eliptik Eğri algoritmaları bilgisayar ağlarından banka işlemlerine, görüntü işleme uygulamalarından mobil platform uygulamalarına, elektronik oy işlemlerinden kablosuz iletişim uygulamalarına kadar birçok alanda kullanılmıştır.

1.3. MOBİL İLETİŞİMDE ASİMETRİK ŞİFRELEME

Mobil aygıtlarda iletişim önemli bir yer tutmaktadır. Gelişen mobil platform donanımları ve mobil cihaz yazılımları sayesinde mobil platformlar üzerinde görüşme yapma ve karşılıklı mesajlaşmanın yanı sıra bankacılık, e- ticaret gibi birçok farklı uygulama hayatımıza girmiştir. Bu uygulamalar kendi gelişmişlikleri ile birlikte belli başlı önemli güvenlik açıklarını da beraberinde getirmiştir.

Araştırma konumuz olan, internet üzerinden karşılıklı mesajlaşmanın güvenilir hale getirilmesi, üçüncü kişilere ve ataklara karşı güvenli bir iletişim yolunun oluşturulması, mesajların şifreli bir şekilde gönderilip alınabilmesini gerektirir. Şifreleme ve şifre çözme işlemleri, şifreleme algoritmasının karmaşıklığına göre değişiklik gösterir. Bu karmaşıklık arttıkça şifreli metnin güvenilirliği artar. Cihazın ram ve batarya kullanımı da bu karmaşıklık yapısına oranla doğrusal olarak artar. Fakat mobil cihazlar bataryalarla çalıştıklarından güç kapasiteleri kısıtlıdır [2]. Bu durumda cihaz üzerindeki bu tür uygulamaların verimli çalışabilmeleri gerekir. Burada yapılması hedeflenen karşılıklı mesajlaşmada şifreleme ve şifre çözme algoritmalarının optimum düzeyde çalıştırılması sağlanarak, hem uygulama güvenilirliği ve performansı elde etmek hem de batarya güç tüketimini minimuma indirmektir.

Bu yol sayesinde ortamda bulunan üçüncü kişiler haberleşen iki nokta arasında gidip gelen verileri elde etseler bile anlamlandıramazlar. Şekil 1.1'de bir örneği verilen bu yol mobil bir cihaz yâda bilgisayar ağlarında oluşan bir ortam olabilir.



Şekil 1.1. Güvensiz Bir Ortamda Şifreli Haberleşme.

Göndericinin yolladığı veriye açık veri denir. Açık veri şifre bloğundan geçip şifreli halini alır. Ardından güvensiz haberleşme ortamından geçen şifreli verinin üçüncü kişiler tarafından ele geçirilmesi hiçbir şey ifade etmez. Şifreli veri alıcıya ulaştığında sadece alıcının bildiği parametrelerle çözüldükten sonra anlamlı hale gelir [5].

Günümüz şifreleme sistemleri açık anahtar ve gizli anahtar şifreleme sistemlerinin birleşiminden oluşmaktadır. Genel olarak açık anahtar şifreleme sistemi, anahtar kurulumu ve sayısal imzayla kimlik denetimi amacıyla kullanılmaktadırlar. Şifreleme işlemleri ise açık anahtar mekanizmasının oluşturduğu anahtarlarla gizli anahtar algoritmaları ile yüksek hızda şifreleme işlemlerini gerçekleştirmektedir [3].

1.4. ŞİFRELEMENİN TARİHÇESİ

Yapılan araştırmalar sonucunda bilgiyi gizleme ve koruma olayının yaklaşık 4000 yılık bir geçmişe sahip olduğu ortaya çıkmıştır.

M.Ö. 1900'lü yıllarda Mısırlı bir kâtibin yazdığı kitabelerde sıra dışı hiyerogliflerin kullanıldığı görülmüştür. Yine Mısırlılardan sonra Mezopotamyalılar ve İbranilerin de bazı metinleri benzer şekillerde kodlamışlardır.

M.Ö 400'lü yıllarda Spartalılar tarafından "scytale" isimli bir sistem geliştirilmiştir. Bu sistemde öncelikle uzun bir parşömen yâda papirüs silindirik bir sopaya sarılıyordu. Ardından gizlenecek metnin kelimeleri sopa üzerinde uzunlamasına her şeride bir harf gelecek şekilde yazılıyordu. Daha sonra parşömen gönderilmek üzere çıkartıldığında, üzerinde anlamsız harflerden oluşan bir metin ortaya çıkıyordu.

Mesajın çözülmesi için gerekli şart şifreleme işleminde kullanılan silindir ile aynı çapta olan bir silindirin kullanılmasıydı. Çünkü farklı çaplardaki silindirlerde yine anlamsız metinler ortaya çıkıyordu.

Tarihteki bir başka teknik ise, MÖ.60-50 yılları arasında Julius Caesar'ın (MÖ 100-44) alfabedeki harflerin yerlerini değiştirerek oluşturduğu şifreleme yöntemidir. Bu yöntemi devlet haberleşmesinde kullanmıştır. Bu yöntem açık metindeki her harfin alfabede kendisinden üç harf sonra gelen harfle yer değiştirilmesine dayanıyordu. Yerine koymalı şifrelemenin basit yöntemlerinden birisin olan bu yöntem daha sonra geliştirilerek monoalfabetik yerine koymalı şifre yöntemi ortaya çıkmıştır.

Monoalfabetik yerine koymalı şifre yönteminde alfabedeki her harfin yerine gönderici ve alıcının bildiği bir başka harf konuluyordu. Caesar şifre yönteminden farkı bir düzene bağlı olmamasıydı.

Arap bilim adamı Al-Kindi'nin 1987 de İstanbul'da bulunan el yazmasından monoalfabetik yerine koymalı şifrenin Araplar tarafından kırıldığını bulması onların bu yöntem konusundaki bilgisini göstermiştir. Bu yöntemin şifresinin çözülmesi yeni şifrelerin geliştirilmesine olanak sağlamıştır. Bunun sonucu olarak her harfin, ortak kelimelerin ve boşlukların yerine birden fazla sembol geliştirilmeye başlanmıştır. Bu şifrenin kayıtlara geçen örnekleri İngiltere kraliçesi I. Elizabeth'e yapılacak suikast planları ve 17. yy da Fransa'da XIV Louis'in Büyük Şifresidir (Great Cipher). Louis'in bu şifresi, 2 yy boyunca kumandan Etienne Bazeries tarafından kırılana kadar çözülememiştir.

20. yüzyılda ise kriptoloji alanı hayati öneme sahip olmuştur. Kriptoloji 2.Dünya Savaşının kaderini belirleyecek seviyeye ilerlemiştir. İttifak devletlerinin ünlü Alman "Enigma" ve Japonların "Purple" kodlarını kırmaları II. Dünya savaşının sonucunu belirleyen en önemli faktörlerden birisidir.

IBM tarafından 1970'li yıllarda çalışmalarına başlanan ve 1976 yılında bilgilerin şifrelenmesi amacıyla ABD'nin federal bilgi işleme standardı olarak belirlenen veri şifreleme standardı (Data Encrypting Standard- DES) tarihteki en fazla kullanıma sahip kriptografi mekanizmalarından biridir.

Kriptoloji alanındaki çarpıcı gelişmelerden bir diğeri, 1976 yılında Helman ve Diffie tarafından yayınlanan "New Directions in Cryptography" bildirisiyle gerçekleşmiştir. Bu bildiri, genel-anahtar kriptografi terimi ile beraber, anahtar alışverişi için yeni bir metot ortaya koymuştur. 1978'de Adleman, Rivest ve Shamir tarafından günümüzde RSA olarak bilinen ilk genel anahtar şifreleme ve imzalama yapısı geliştirilmiştir. RSA'nın yapısı, büyük tamsayıların çarpanlara ayrılması temeline dayanmaktadır.

1985 yılında Neal Koblitz ve Victor S.Miller birbirlerinden ayrı yaptıkları çalışmalarda eliptik eğri (ECC) sistemlerini tarif etmişlerdir. 1990 yılında James Massey ve Xuejia Lai IDEA algoritmasını, 1991 yılında Phil Zimmerman PGP sistemini geliştirmiş ve yayınlamıştır. 1995 yılında ise NIST tarafından SHA-1 (Secure Hash Algorithm) özet algortiması standart olarak yayınlanmıştır.

1997 yılında ABD'nin NIST (National Institute of Standards and Technology) kurumu DES'in yerini alması için bir simetrik algoritma yarışması başlatmış, bu yarışmayı 2001 yılında Belçikalı Vincent Rijmen ve Joan Daemen'e ait Rijndael algoritması kazanmış ve bu algoritma AES (Advanced Encryption Standard) adıyla standart haline getirilmiştir.

2005 yılında Çin'li bir ekip, SHA-1 algoritmasının kırıldığını duyurmuştur. Buna göre 2^{80} gücündeki algoritma, 2^{63}'e kadar indirilmiştir. Bu durum üzerine Amerikan Hükümeti ve Sun, Microsoft gibi birçok büyük firma bu algoritmayı artık kullanmayacaklarını açıklamışlardır.

2. MATERYAL VE YÖNTEM

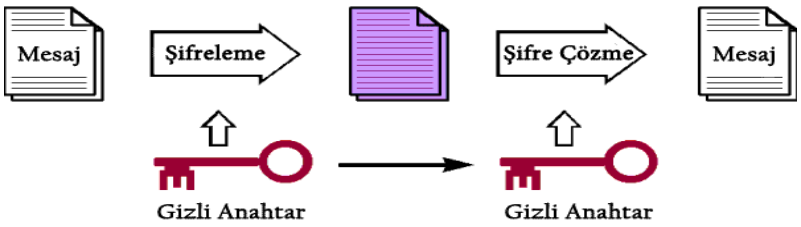
2.1. ŞİFRELEME TÜRLERİ

Kriptografide Gizli Anahtarlı Şifreleme (Simetrik) ve Açık Anahtarlı Şifreleme (Asimetrik) olmak üzere iki adet şifreleme türü vardır.

Simetrik şifrelemede şifrelenerek gönderilmek istenen veri bir anahtar tarafından algoritmaya sokulur ve şifrelenir. Şifrelenmiş veri alıcıya gönderilir. Şifreli veri kendisi ulaşan alıcı taraf ise, şifreli metni eski haline çevirebilmek için yine aynı anahtarı kullanır. Yani simetrik şifrelemede kriptolamak ve çözmek için kullanılan anahtarlar aynıdır [4]. Asimetrik şifrelemede ise mesajı şifrelerken kullanılan anahtar ile çözmek için kullanılan anahtar birbirinden farklıdır. Şifrelemek için "açık anahtar" çözmek için ise "gizli anahtar" kullanılır. Örneğin mesajı X şifresiyle şifreleyip gönderdiğimizde, mesaj alıcı tarafta sadece Y anahtarıyla çözülür. Yine tam tersi yönde Y anahtarıyla şifrelenen metin yalnızca X anahtarıyla çözülebilir.

2.1.1. Gizli Anahtarlı Şifreleme (Simetrik)

Şekil 2.1'de blok şeması verilmiş olan gizli anahtar şifreleme sisteminde, bir anahtar hem mesajın şifrelenmesi hem de şifresinin çözülmesi işleminde kullanılır [4]. Bu nedenle sistem asimetrik sisteme göre daha hızlı çalışır.



Şekil 2.1. Gizli Anahtarlı Şifreleme.

Sistem, şifrelenecek mesajın aynı ortam içerisinde kaldığı ve üçüncü kişilerin mesaja saldırı yapma ihtimalinin olmadığı ortamlarda oldukça kullanışlıdır. Mesajın farklı ortamlara taşınması gerektiği durumlarda, mesaj anahtarının da güvenli bir şekilde karşı tarafa taşınmasını gerekir. Bu durum ise güvenlik zaafiyetine neden olur.

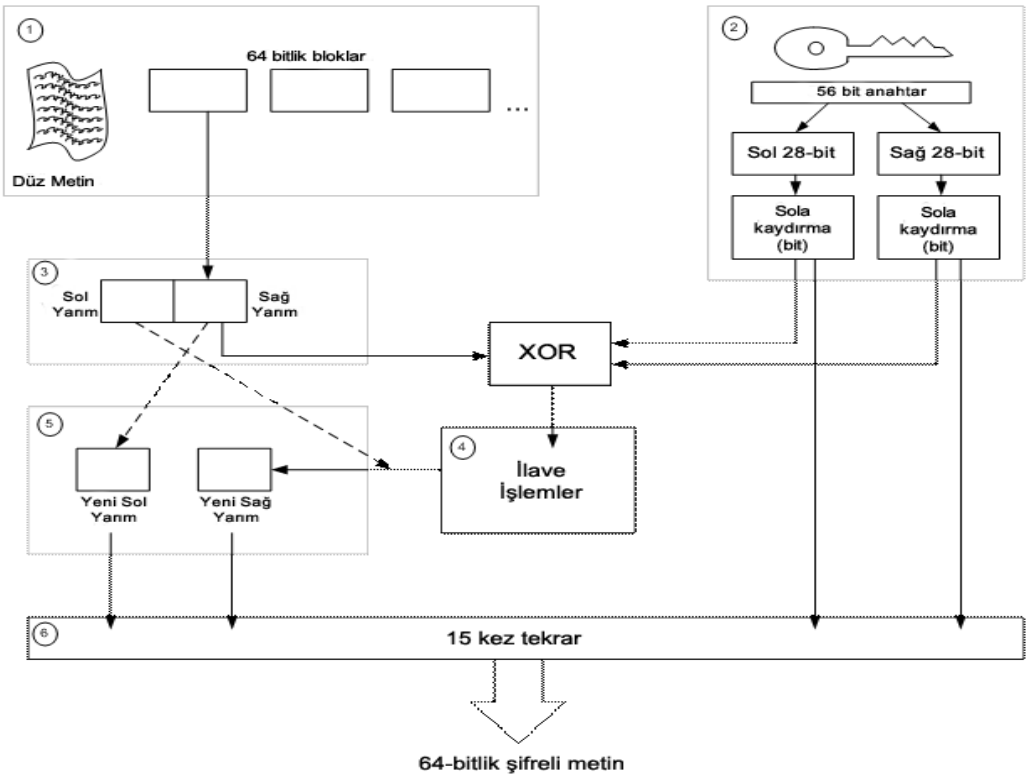
Anahtarı elinde bulunduran birisi şifreli veriyi kolaylıkla çözebilir ve mesaj içeriğine ulaşabilir. Bu nedenle mesajı şifreleyip iletim ortamına bırakan kişi ile şifreli veriyi alıp, bu veriyi anlamlı mesaj haline çevirecek olan kişinin anahtar üzerinde anlaşma sağlamış olmaları, aracı bir kurye veya güvenli bir iletişim şekli ile anahtar dağıtımını gerçekleştirmeleri ve anahtarı kendi aralarında kullanıp, başkalarına karşı gizli tutmaları gerekir [3]. Sistemin güvenliği algoritmanın gizliliğinden değil, anahtarın gizliliğinden gelmektedir. Simetrik anahtar kriptolamaya, Rijndael, Blowfish, IDEA (International Data Encryption Algorithm), DES (Data Encryption Standard) ve AES (Advanced Encryption Standard) algoritmaları örnek olarak verilebilir.

Çalışmanın bu kısmında DES ve AES algoritmalarının yapıları incelenmiştir.

2.1.1.1. DES (Data Encryption Standard)

Açılımı Data Encryption Standard olan simetrik şifreleme algoritmasıdır. DES algoritması, geniş kullanım alanına sahip bir blok kriptolama algoritmasıdır. Büyük boyuttaki verilerin şifrelenmesi işlemlerinde kullanılır. Şekil 2.2'de blok şeması verilen ve blok şifreleme olarak isimlendirilen bir yöntem kullanılarak gerçekleştirilir. Bu yöntemde mesaj öncelikle belirli uzunlukta bloklara bölünür. Her blok ayrı ayrı şifrelenir ve en sonunda birleştirilerek şifreli bir metin yapısı elde edilir [6].

Algoritmada her blok 8 bit parity biti olmak üzere 64 bit uzunluğundadır. Kullanılan işlemci sayısına göre blok uzunluğu değişebilir. Son dönem bilgisayarlarda, 128 bit uzunluğu kullanılmaya başlanmıştır.



Şekil 2.2. DES Algoritma Yapısı.

DES kaba kuvvet saldırlarına karşı güvensizdir. Bu nedenle DES'in güvenilirliğini arttırmak amacıyla 3DES metodu geliştirilmiştir. Bu metotta, şifrelenen veri yeniden geri çözülür ve DES şifrelemesi 3 kez art arda uygulanır. Şifreleme işlemi için 24 byte uzunluğundaki anahtar 3 bloğa ayrılır.

Anahtarın ilk 8 byte'ı ile DES şifrelemesi yapılır. Ardından şifrelenen metin ikinci bloktaki 8 byte ile çözülür ardından son 8 byte ile yeniden şifrelenerek 8 byte'lık blok elde edilir. Yeni yöntemde DES algoritmasının güvenilirliği artarken, hızı 3 kat oranda azalmıştır.

DES algoritmasını kırmak için yüksek maliyetli son teknolojik cihazlar geliştirilmiş olmasına rağmen, başta devlet daireleri ve bankalar olmak üzere birçok ortamda 3DES algoritması kullanılmaya devam etmektedir.

Algoritmanın kırılma tehdidini gören Ulusal Standartlar Enstitüsü (NIST), 2000 yılında DES algoritmasını AES (Advanced Encryption Standard; Gelişmiş Şifreleme Standardı) ile değiştirmiştir [7].

Bilinen Saldırılar

- *Brute Force:* Saldıran kişi özel bir program yardımıyla tüm olasılıkları deneyerek anahtarı tahmin etmeye çalışır.
- *Differansiyel Kriptanaliz Atak:* Açık metin ataktır ve saldırı mantığı S kutularının düzgün olmayan diferansiyel dağıtım tablolarına dayanır.
- *Lineer Kriptanaliz Atak:* Differansiyel kriptanaliz atağına göre algoritma bu saldırıya karşı daha savunmasızdır.

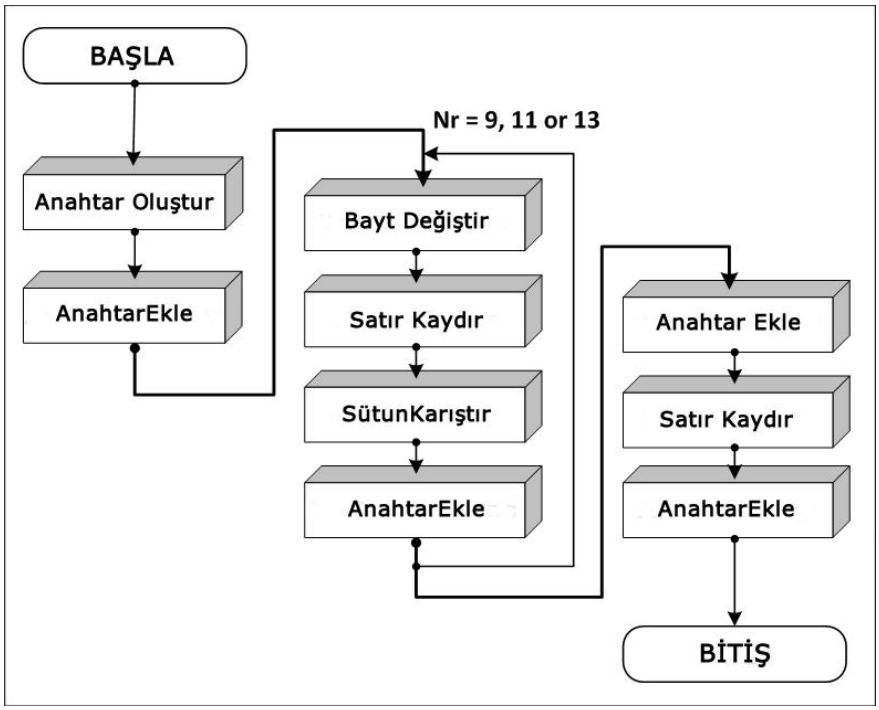
2.1.1.2. AES (Advanced Encryption Standard)

DES algoritmasının gelişen teknoloji ve artan işlemci hızları karşısında güvenilirliğini yitirmesi üzerine 1997 yılında NITS tarafından, yeni bir şifreleme standardını belirlemek amacıyla bir yarışma düzenlenmiş, düzenlenen yarışma sonucu 2000 yılında, Vincent Rijmen ve Joan Daemen tarafından geliştirilen Rijndael algoritması üzerinde standartlaşma ve düzenlemelere gidilerek, bu algoritmanın Gelişmiş Şifreleme Standardı (AES) olarak kullanılacağı duyurulmuştur.

Algoritmanın Genel Yapısı

Algoritma, simetrik-anahtarlı yani, şifreleme ve şifre çözme anahtarları aynı olan, yazılım ve donanım performansı yüksek, ram gereksinimi düşük bir algoritmadır. 128-bit girdi bloğu, 128, 192 ve 256 bit anahtar uzunluklarına sahiptir. Algoritmada giriş, çıkış ve matrisler 128 bitliktir.

Yapı, durum (state) denilen 4 satır, 4 sütun yani 4x4 sütun-öncelikli bayt matrisi üzerinde çalışır. Matristeki işlemler sonlu özel bir cisim (finite field) üzerinde yapılmaktadır. 16 bölümden oluşan durumun her bölümüne bir baytlık veri düşer. Her satırda 32 bit uzunluğunda bir kelime ortaya çıkar. Şekil 2.3'de algoritmanın genel yapısı verilmiştir.



Şekil 2.3. AES Algoritma Yapısı.

Algoritma içerisine giren açık metin belirli sayıda döngü işlemine tabi tutularak şifreli çıktı metni elde edilir. Döngü sayısı anahtar uzunluğuna göre değişiklik gösterir. 128-bit için 10, 192-bit için 12, 256-bit için 14 defa döngü işlemi uygulanır.

Her bir döngüde 4 ayrı alt işlem gerçekleştirilir. Bunlar sırasıyla bayt değiştirme, satır kaydırma, sütun karıştırma ve tur anahtarı ile toplamadır. Tüm döngüler sona erdiğinde giren verinin şifrelenmiş hali dışarı çıkmaktadır. İlk döngüde anahtar ilk haliyle katılmakta diğer döngülerde ise yeni üretilen anahtarlar sokulmaktadır [8].

Şifreli metni geri çözmek için ise, bu döngüler ters sıra ile uygulanarak açık metin elde edilir.

Bilinen Saldırılar

AES algoritmasına yönelik bilinen en önemli saldırı yöntemi XSL ataktır. Saldırı şifreli verinin analizi ve quadatik eş zamanlı eşitlikleri elde etmek üzere kuruludur. Bu eşitlikler çok geniştir. Örneğin 128 bitlik AES için bu sayı 1600 değişken ve 8000 eşitliktir.

Tam çevirim AES algoritmasına yapılan ilk başarılı anahtar elde etme saldırısı Christian

Rechberger, Dmitry Khovratovich ve Andrey Bogdanov tarafından gerçekleştirilmiştir. Saldırıda biclique adı verilen yapılar kullanarak 128-bit anahtarı $2^{126.1}$, 192-bit anahtarı $2^{2189.7}$ ve 256-bit anahtarı $2^{254.4}$ işlem karmaşıklığıyla elde etmektedir.

Yan kanal saldırılarında ise, 2005 yılında D.J. Bernstein OpenSSL'in AES uygulamasını kullanan bir sunucuya ön-bellek-zamanlama saldırısı yapmıştır. Saldırıda algoritmanın bulunduğu sunucuya 200 milyon civarında açık metin gönderilmiş ve şifreleme işleminin kaç saat sürdüğü bilgileri elde edilmiştir. Bernstein, bu saldırının başarıya ulaşması için, şifreleme süresinin sunucudan gelen kadar kesin olmasına gerek olmadığını göstermiştir.

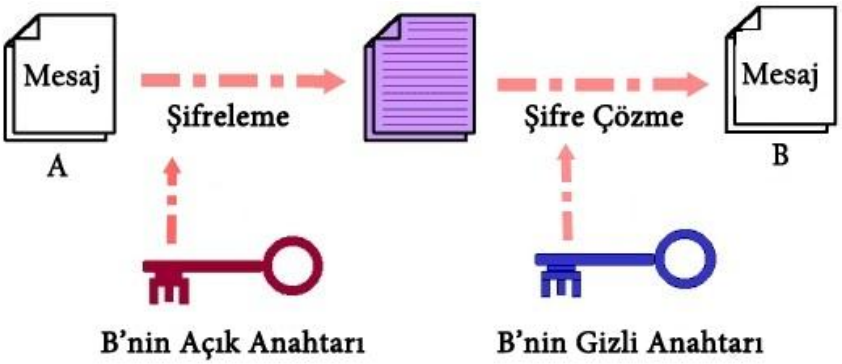
Aynı yıl Eran Tromer, Adi Shamir ve Dag Arne Osvik yine bir ön bellek-zamanlama saldırıları ile AES anahtarını 65 milisaniye içerisinde elde edebilen bir yöntem geliştirmişlerdir. Bu saldırının başarılı olma koşulu, saldıran kişinin saldırdığı platformda AES yöntemi ile şifreleme yapabilmesi yani kod çalıştırabilmesi gerekiyordu.

2009'da ise, FPGA kullanan bir platform üzerinde, 2^{32} zorlukla algoritma anahtar elde edilmiştir.

Stephan Krenn, David Gullasch ve Endre Bangerter, 2010 yılında açık veya şifreli metin kullanmadan OpenSSL gibi sıkıştırma tabloları kullanan uygulamalardan AES-128 gizli anahtarını elde etmeyi sağlayan bir saldırı yayımlamışlardır. Bu saldırıda da saldıran kişinin, saldırdığı platformda kod çalıştırabilmesi gerekiyordu [9].

2.1.2. Açık Anahtarlı Şifreleme (Asimetrik)

Açık anahtarlı şifreleme, şifreleme ve şifre çözme yöntemlerinde farklı anahtar yapılarının kullanıldığı bir şifreleme sistemidir. Şekil 2.4'de blok şeması verilen ve simetrik şifreleme sistemi olarak da bilinen bu kriptografi sisteminde, açık ve gizli anahtar olarak adlandırılan bir anahtar çifti kullanılmaktadır [4]. Anahtar çiftlerini üreten algoritma yapılarının matematiksel özelliklerinden dolayı açık-gizli anahtar çiftleri her kişi için farklılık gösterir. Yani her kullanıcının açık-gizli anahtar çifti sadece o kullanıcıya aittir. Bu anahtarlardan hiçbiri hem şifreleme hem de deşifreleme işlemi için kullanılmaz.

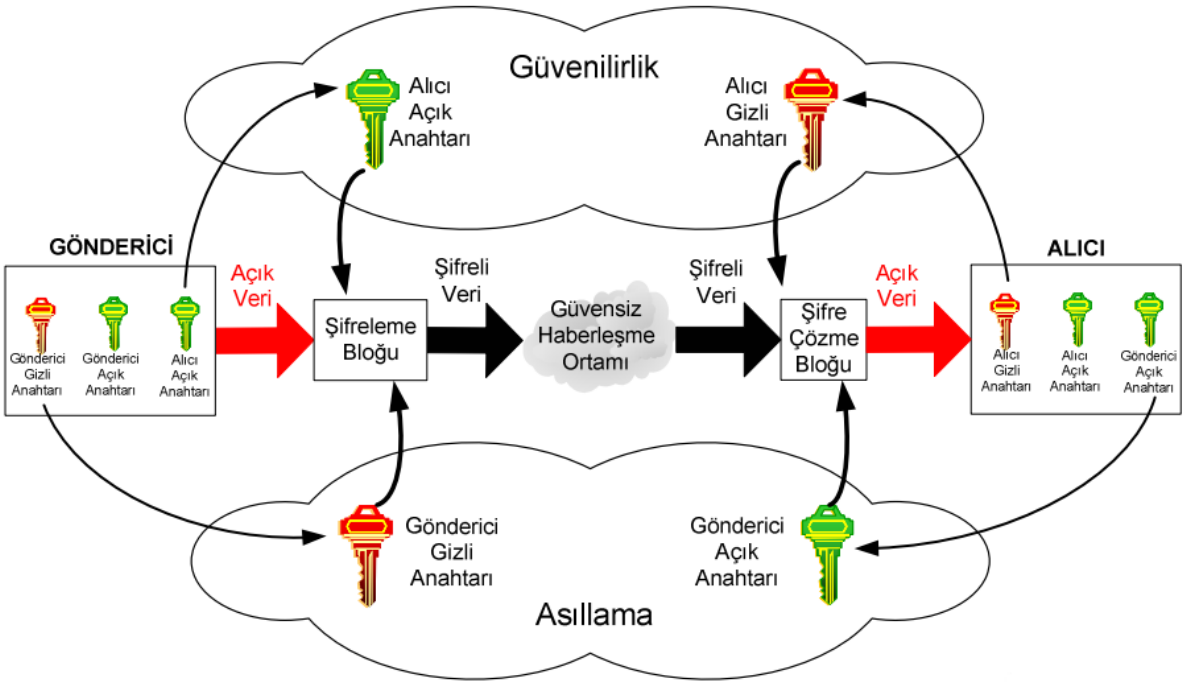


Şekil 2.4. Açık Anahtarlı Şifreleme.

Gizli anahtarın sadece bir sahibi vardır. Kişi gizli anahtarı aracılığıyla, kendi açık anahtarıyla şifrelenmiş olan bilgilerin şifresini çözebilir, kendisine ait sayısal imzalar üretebilir ya da kendi kimliğini ispat edebilir [10].
Açık anahtar ise yanlızca gizli anahtarın sahibi tarafından oluşturulabilir. Anahtar, elektronik kimlik belgeleri içerisinde diğer kişisel bilgiler ile birlikte tutulur ve herkes birbirinin açık anahtarını istedikleri zaman elektronik kimliklerine ulaşmak suretiyle elde edebilir [11].

Açık anahtar sistemleri iki amaçla kullanılabilir. Bunlar güvenilirlik ve kimlik doğrulama(asıllama)dır. Yabancı bir kişi bir iletiyi şifrelemek istediğinde, güvenilirlik için bu iletiyi göndermek istediği kişinin şifreleme anahtarını (açık anahtarı) kullanır. Şifrelenmiş veriyi ancak ilgili şifre çözüm anahtarına (gizli anahtara) sahip olan kişi çözebilir. Eğer şifreli veri açılamazsa haberleşme esnasında bozulmuş olarak kabul edilir. Anahtar yapıları birbirlerinden farklı olsalar bile, matematiksel olarak birbirleriyle ilişki halindedirler. Şekil 2.5'de açık anahtar şifreleme sisteminin blok şeması verilmiştir.

Karşılıklı haberleşmek isteyen iki kişi asıllama için açık anahtar şifreleme sistemini kullanacaksa bu durumda iletiyi imzalayan gönderici kişi kendi gizli anahtarını kullanır. Alıcı, imzayı göndericinin açık anahtarı ile doğrular. Eğer başarabilirse ileti o açık anahtara ait gönderici tarafından imzalanmıştır, göndericiyi asıllamış olur.



Şekil 2.5. Açık Anahtar Kriptografisi.

Genel olarak açık anahtarlı şifreleme sistemleri üç ana başlık altında incelenir [12];

- *Şifreleme ve Şifre Çözme:* Gönderici veriyi alıcının açık anahtarı ile şifreler.
- *Sayısal İmza Uygulaması:* Gönderi özel anahtarı ile mesaja ait imza üretir. Bu imza mesajın tamamı ya da bir kısmı için yapılabilir.
- *Anahtar Değişimi:* Gönderici ile alıcı haberleşme için kurulan oturumun anahtarlarını değiş tokuş ederler.

Uygulamada açık anahtar algoritmaları 3 gruba ayrılır [4];

- *Bir Tamsayının Çarpanlarına Ayrılmasına Dayanan Algoritmalar: Verilen* bir n pozitif tamsayının asal çarpanlarını bulmaya dayanır. n sayısı çok büyük ve özel seçilmektedir. RSA[1] en çok kullanılan açık anahtar algoritması olup bu problemin çözümünün zorluğuna dayanmaktadır.
- *Ayrık Logaritma Problemine Dayanan Algoritmalar:* Verilen bir α , β ve p için öyle bir x değeri bulunmalıdır ki $\beta = \alpha^x \bmod p$ sağlamalıdır. Diffie-Hellman anahtar değişim protokolü, ElGamal algoritması bu probleme dayanmaktadır.

- *Eliptik Eğri Ayrık Logaritma Problemine Dayanan Algoritmalar:* Bu algoritmalar da Ayrık Logaritma Problemi tabanlıdır ancak burada Ayrık Logaritma Problemi bir eliptik eğri üzerinde tanımlıdır.

  Eliptik eğri Diffie-Hellman protokolü ve eliptik eğri sayısal imza algoritması bu matematiksel temele dayanır. Diğer grupta yer alan algoritmalarla 1024-2048 bit uzunluğunda sağlanan güvenlik seviyesi 160-256 bit uzunluğu ile sağlanabilmektedir. Sonlu alan aritmetiğine dayanmaktadır [5].

2.1.2.1. Avantajları

Sistem, simetrik şifreleme algoritmalarının aksine "anahtar değişimi" ihtiyacına gerek duymaz. Çünkü simetrik sisteme bakacak olursak; sistemin çalışmasında, kriptolamak ve çözmek için kullanılan anahtarın hem mesajı gönderen hem de alan tarafta bulunması gerekir. Bunun sonucu olarak bu anahtarın gönderici taraftan alıcı tarafa güvenli olarak iletilmesi gerekir. Bu durum simetrik sistemlerde güvenlik sorununu beraberinde getirir iken asimetrik sistemlerde böyle bir durum söz konusu değildir. Çünkü asimetrik şifrelemede mesajı şifreleyen tarafın kullandığı anahtarın çözücü eşi alıcı tarafta bulunmalıdır.

Bu sistemde anahtarlardan birine “açık anahtar” (yani herkes tarafından görülebilen anahtar) denirken diğerine ise “gizli anahtar” (sadece taraflardan birine ait olan anahtar) denir. Gizli anahtar yalnızca bir tarafa aittir. Yani alıcıya tahsis edilen gizli anahtar yalnızca alıcı için oluşturulmuş ve yalnızca alıcı tarafından kullanılabilen bir anahtardır. Bu anahtarın eşi olan anahtar ise *“Anahtar kütüphanelerinde”* tutulur.

Şifreli bir mesaj gönderecek olan kişinin bu kütüphaneye gidip alıcının açık anahtarını kullanarak göndereceği mesajı şifrelemesi gerekir. Şifreli metin alıcıya ulaştığında alıcının yapması gereken tek şey gizli anahtarıyla metni şifresiz formuna dönüştürmek olacaktır.

Sistemin bir diğer avantajı ise, simetrik sistemde algoritma anahtarı aynı olması gerektiğinden ve birden fazla kişide aynı anahtar bulunduğundan şifreme işlemini yapan kişinin ayırt edilememesidir. Şifreleme işlemini yapan kişi bir müddet sonra bu işlemi kendisinin yaptığını inkâr edebilir. Aksi kanıtlanamaz çünkü aynı anahtar şifreli metni çözecek kişinin elinde de mevcuttur. Buna karşılık asimetrik şifreleme sistemlerinde

kullanıcıların yanlızca kendi anahtarlarını gizli tutması yeterlidir.

2.1.2.2. Dezavantajları

Asimetrik şifreleme sistemi, simetrik şifreleme sistemi ile karşılaştırıldığında daha yavaştır. Bu durumun nedeni asimetrik şifreleme işleminde kullanılan anahtar uzunluğunun simetrik sistemdeki anahtara göre daha uzun olması ve bu nedenle yapılan işlemlerin daha karmaşık olmasıdır. Bunun sonucu olarak açık anahtar sisteminin, performans gerektiren şifreleme uygulamalarında (örneğin yoğun bir ağda çalışan bir ağ güvenlik cihazında) şifreleme amacıyla kullanılması mümkün değildir [13].

Bir diğer sorun, simetrik şifreleme sisteminde bir mesajın aynı anda birden fazla kişiye gönderilmesi gerektiği durumlarda, her kullanıcı için ayrı ayrı şifrelenmesi gerektiğidir.

Bir diğer sorun ise, asimetrik sistemlerin güvenliğinin henüz ispatlanmamış varsayımlara dayanmasıdır. Asimetrik şifreleme sistemlerinde güvenlik tek-yönlü fonksiyonlara dayanır. Bu fonksiyonların kendisinin hesaplanması "kolay", tersinin hesaplanması ise *"imkansız"'*dır. İmkânsızdan kastedilen, fonksiyonun tersinin hesaplanabilmesinin polinomial süre içerisinde imkânsız olmasıdır. Ancak ters alma işlemini hızlandıracak yöntemlerin var olmadığı henüz ispatlanmış değildir. [53]

2.1.2.3. Açık Anahtar Altyapısını Kullanan Algoritmalar

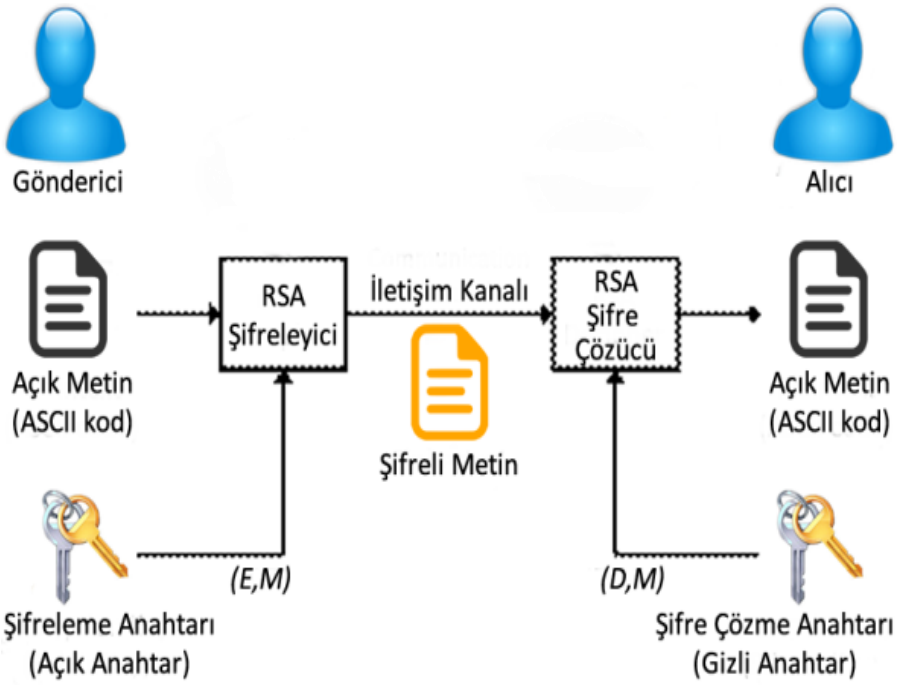
Açık anahtarlama şifreleme yapısını kullanan algoritmalarına bakıldığında Diffie-Hellman, RSA [1], Eliptik Eğri [2], ElGamal, Paillier gibi algoritmaların mevcut olduğu görülür. Tezimizde kullandığımız RSA ve Eliptik Eğri sistemlerinin algoritma yapıları aşağıda verilmiştir.

2.1.2.3.1. Rsa Şifreleme Algoritması

RSA[1] şifreleme algoritması, dijital ortamda verilerin güvenli bir şekilde aktarılmasının sağlanması fikri temel alınarak, tam sayıları çarpanlarına ayırmanın algoritmik zorluğuna dayanan bir tür açık anahtarlı şifreleme yöntemidir. Günümüzde en çok kullanılan hem sayısal imza atma hem de şifreleme olanağı tanıyan yöntem olarak bilinir. 1978'de Adi Shamir, Ron Rivest ve Leonard Adleman tarafından ortaya

çıkarılmıştır.

Şekil 2.6'da algoritma yapısı verilen RSA şifreleme yönteminde, anahtar oluşturma işlemi içerisinde asal sayılar kullanılır. Bu da daha güvenli bir yapı oluşturulmasını sağlar.



Şekil 2.6. RSA Algoritma Yapısı.

Algoritmanın Yapısı [54]

• P ve Q gibi çok büyük iki asal sayı seçilir.

• Bu iki asal sayının çarpımı N = P.Q ve bir eksiklerinin φ(N)=(P-1)(Q-1) değeri hesaplanır.

• 1'den büyük φ(N)'den küçük φ(N) ile aralarında asal bir M tamsayısı seçilir.

• Gizli üs D, seçilen M tamsayısının mod φ(N)'de tersi alınarak elde edilir.

• M ve N tamsayıları genel anahtarı, D ve N tamsayıları ise özel anahtarı oluşturur. P,Q ve φ(N) değerleri de özel anahtar gibi gizli tutulmalıdır.

Genel ve özel anahtarlar oluşturulduktan sonra iletilmek istenen mesaj genel anahtar ile

şifrelenir. Şifreleme işlemi şu şekildedir: Şifrelenecek mesajın sayısal karşılığının M' ninci kuvveti hesaplanır ve bunun mod N deki karşılığı şifrelenmiş metni oluşturur.

Genel anahtar ile şifrelenmiş olan metin ancak özel anahtar kullanarak açılabilir. Bu nedenle şifrelenmiş metin, yine aynı yolla, şifrelenmiş metnin sayısal karşılığının D'ninci kuvveti alınıp, bunun mod N deki karşılığı bulunarak orijinal haline çevrilebilir.

RSA Algoritmasına Yapılan Ataklar

Yapılan saldırılar yan kanal ve kriptanaliz olmak üzere ikiye ayrılır. Kriptanaliz, algoritma üzerine uygulanır ve algoritma zayıflığından yararlanmaya çalışır. Yan kanal saldırıları ise algoritmanın çalıştığı elektriksel ve fiziksel ortama uygulanır ve ortamın sağladığı bilgilerden yararlanıp şifreyi çözmeye çalışır. RSA'ya yapılan yan kanal analizi ataklarında başarılı sonuçlar alınmış olmasına rağmen kriptanaliz yöntemleri ile hala RSA'ın kırılması gerçekleşmemiştir [15]. Algoritmanın kırılması için eğer yeni saldırı metotları veya yeni güçlü makineler varsa, yapılması gereken şey algoritma içerisinde daha büyük asal sayılar kullanmaktır. Asal sayıların büyüklüğü sonucunda, her bir saldırı daha ağır matematiksel işlemler doğurur. Bu durum ise algoritmanın gücünü arttırır.

Kriptanaliz saldırlarda bilinen;

- *Faktörizasyon Atak:* Giriş metni bozmak için çeşitli faktörizasyon algoritmaları vardır. RSA bunu yenmek için n parametresinin 300 decimal bit değerinden daha fazla olmasını ister. Yani modülün minimum 1024 bit olması istenir [16].
- *Lattice Tabanlı Atak:* Lattice indirgeme algoritmaları ile parametreleri bulmaya çalışır [17].

Algoritmaya uygulanan yan kanal saldırıları aşağıda verilmiştir.

Zamanlama Saldırıları

Algoritmanın veya algoritma adımlarının her biri üzerinde gerçekleştirilen işlemlerin ne kadar sürede çalıştığını tespit etmeye ve bu tespite göre çıkarım yapmaya çalışır. Bu alanda ilk çalışma 1995 yılında Paul Carl tarafından yapılmıştır [18]. Bu çalışmadan 5 yıl sonra Werner Schindler Çinli kalan teoremi ile RSA'a karşı bir zamanlama saldırısı yapılabileceğini göstermiştir [19].

Güç Analizi Saldırıları

Güç analizi saldırılarında transistor üzerinden geçen akım farklarından yararlanılarak bilgi elde edilmeye çalışılır. Bu saldırı yönteminin iki farklı çeşidi vardır; İşlemcinin harcadığı enerji miktarının analiz edildiği Basit Güç Analizi (Simple Power Analysis) ve istatistiki analizler ve hata düzeltme algoritmaları kullanılarak yapılan Farksal Güç Analizi (Differential Power Analysis) saldırılarıdır.

Hata Analizi Saldırıları

Algoritma işletilirken sistem saati, sıcaklık, voltaj, radyasyon, ışık gibi bazı çevresel parametre değerleri değiştirilerek sistem hatalı çalışmaya zorlanır. Hatalı çalışma sonucu elde edilen yanlış çıktılar ile beklenen doğru çıktılar karşılaştırılarak yorumlanır. Bu yolla algoritma ve kriptografik anahtarın koruduğu veriler hakkında bilgi toplanmaya çalışılır. Hata analizi yöntemiyle RSA'ya yapılan ilk saldırı 2010 yılında gerçekleştirilmiştir [20].

Akustik Kriptanaliz

Şifreleme ve şifre çözme işlemleri işlemcilerin iş yükünü arttırır. Bundan dolayı işlemciler insan kulağının duyabildiği ve duyamadığı bir takım sesler çıkarır. Geliştirilen cihazlar aracılığıyla bu seslerin dinlenmesi ve analiz edilip yorumlanması akustik kriptanaliz kapsamına girer. RSA'yı geliştirenlerden biri olan Adi Shamir, 2004 yılında yanlızca işlemciden çıkan sesi kullanarak zamanlama saldırısı yapabileceğini ispatlamış, ardından bir ekip ile birlikte bu konuda araştırmalarını genişletip saldırı tekniğini duyurmuşlardır [21].

Elektromanyetik Alan Saldırıları

İşlemcinin temelini oluşturan transistor yapıları durum geçişlerinde yüksek akım çekerler. Bu akım değişimi ile birlikte elektromanyetik alan oluşur. Saldırı, bu yan kanal bilgisini analiz ederek veri hakkında bilgi toplamayı hedefler.

Çarpanlara Ayırma Saldırısı

Klasik şifre çözme yöntemi olan kaba kuvvet (brute force) saldırısının RSA uyarlamasıdır. Anahtar üretiminde kullanılan N tam sayısını çarpanlarına ayırarak p ve

q asal sayı değerlerini bulmayı amaçlar [22].

Küçük ‘e’ Sayısı Saldırısı

Kullanıcının aynı mesajı birçok kişiye aynı ‘e’ sayısını kullanarak şifreleyip gönderdiği durumlarda ortaya çıkar. Eğer 'e' sayısı küçük bir sayı ise, araya girmeyi başarıp iletişimi dinleyen kişi topladığı veriler üzerinde Çinli kalan teoremini kullanarak şifreli veriden bilgiyi elde etmeye çalışır.

Kuantum Hesaplama Gücü ile Yapılan Saldırılar

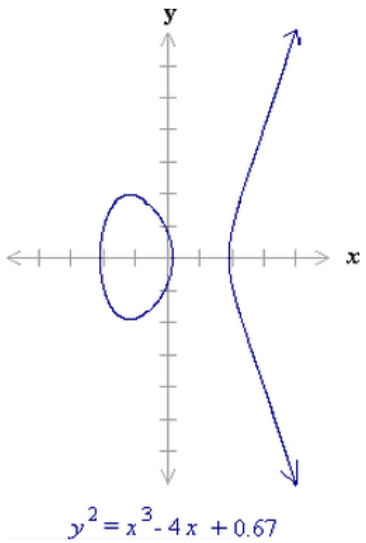
RSA algoritması güvenlik zafiyeti vermemek için büyük asal sayılar ile çalıştığından, algoritma üzerine uygulanan kaba kuvvet saldırıları büyük oranda hesaplama maliyetleri ile karşı karşıya kalırlar. Ayrıca saldırı yapacak sistemlerin ciddi anlamda sağlam bir donanıma ve belki de yıllar sürecek zamana ihtiyacı vardır. Fakat gelecekte hayal edilen kuantum bilgisayarların geliştirilmesi, işlem gücü gerektiren matematik işlemlerin günümüz sistemlere oranla çok daha hızlı yapılmasını sağlayabilecek ve RSA algoritmalarının kısa bir zamanda kırılmasını mümkün hale getirecektir.

2.1.2.3.2. Eliptik Eğri

Eliptik eğriler [2] konusu 19.yüzyıldan beri yoğun olarak çalışılan matematiksel bir konudur.

Eliptik eğri, gerçek sayılar kümesi üzerinde tanımlanan, $y^2 = x^3 + ax + b$ genel denklemini x ve y sayıları için sağlayan eğrinin adıdır. Bu denklem her a ve b değeri için farklı bir eğri sonucu verir.

Örneğin $a = -4$ ve $b = 0.67$ değerleri için $y^2 = x^3 - 4x + 0.67$ denklemi elde edilir. Bu denklemin grafiği Şekil 2.7’de verilmiştir [55].

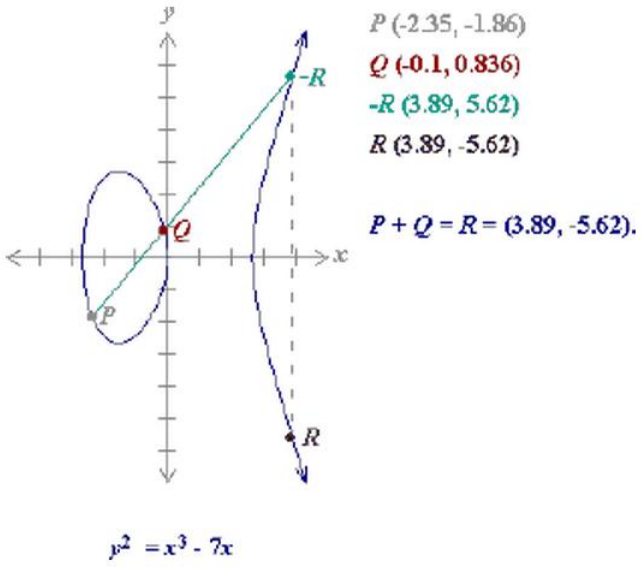


Şekil 2.7. Eliptik Eğri Denklem Grafiği [56].

Şayet $x^3 + ax + b$ denklemin tekrarlı kökü yoksa diğer bir deyişle $4a^3 + 27b^2$ değeri 0 değilse, $y^2 = x^3 - 4x + 0.67$ genel denklemi için bir grup oluşacağı söylenebilir. Eliptik bir grup olarak kastedilen eliptik eğri üzerinde tanımlı olan noktalardır ve bu noktalar öyle bir O noktasında sonsuza gider [23].

Eliptik Eğrilerde Toplama İşlemi

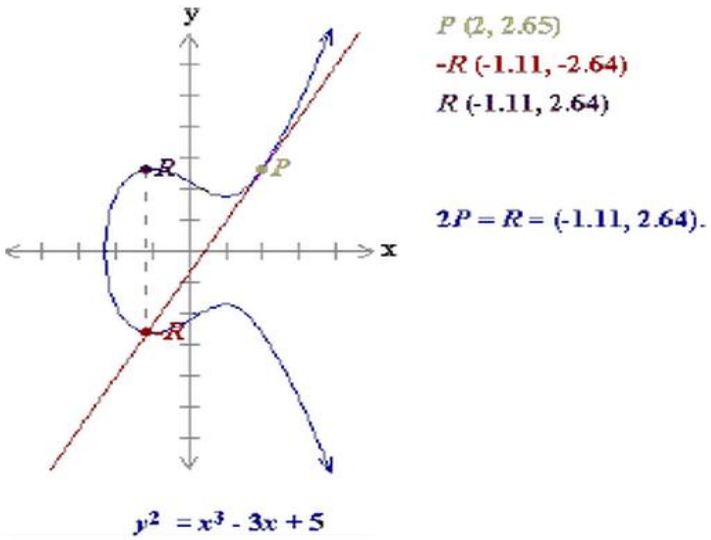
Şekil 2.8'de görüldüğü üzere Eliptik Eğri grupları toplanabilir. Toplama bu grupların en temel fonksiyonudur. Bir Eliptik Eğri üzerinde geometrik olarak iki nokta tanımlanabilir. Örneğin $\mathrm{P} = (\mathrm{xP}, \mathrm{yP})$ noktasını tanımlamak aşağıdaki şekilde olduğu gibi mümkündür. Eliptik Eğrilerin bir diğer özelliği ise x eksenine göre simetrik eğriler oluşudur. Örneğin P noktasının simetriği $\mathrm{P} = (\mathrm{xP}, -\mathrm{yP})$ olarak tanımlanabilir. [55]



Şekil 2.8. Eliptik Eğri Üzerinde Toplama İşlemi [56].

P ve Q eliptik eğri üzerinde iki farklı nokta olsun. Bu iki noktayı toplamak için ilk önce P ve Q noktaları üzerinden geçen bir doğru çizilir (uzayda iki nokta bir doğru belirtir ve burada $Q \neq -P$ olmalıdır çünkü simetrik bir nokta alınması durumunda çizilen doğru y eksenine paralel olur). Bu doğru eğri üzerinde sadece $-R$ adı verilen noktaları kesmektedir. $-R$ noktasının x eksinine göre tersi bize R noktasını verir. Bu nokta P ve Q noktasının toplamıdır [5].

Bir noktanın kendisi ile toplanması da eliptik eğrilerde mümkündür. Örneğin, Şekil 2.9'daki eğride P noktası kendisi ile toplanmıştır [23].



Şekil 2.9. P + P = R [56].

Bir noktanın kendisiyle toplamı o noktadaki eğimin yönünde bir doğrunun, yine eliptik eğri üzerindeki kestiği noktanın x eksenine göre tersi alınarak hesaplanır. P noktasının kendisi ile toplanması esnasında dikkat edilirse P noktasının y değerinin 0'dan farklı olduğu görülür. Fakat bir noktanın y değeri 0 olsa bile kendisi ile toplanması mümkündür. Bu özel durumda noktanın eğimi y eksenine paralel olacağı için eliptik eğriyi ikinci bir noktadan kesmez. Bu durumda P noktasının kendisi ile toplamı O (sonsuz) olacaktır [55].

O değerine P noktası eklenecek olursa sonuç yine P noktasına eşit olur. Bu durumda $5P = P, 6P = O, 7P = P$ olduğunu söylemek doğrudur [23].

Eliptik Eğrilerin Şifrelemede Kullanımı

Eliptik eğri sistemi gerçek sayılar kümesinde çalışır. Bu nedenle şifreleme ve veri güvenliğinde, yuvarlama yâda belirsizlik durumları olmadan tam değer vermesi yönünden dolayı kullanışlıdır.

Bir F_p grubu oluşturulurken 0 ile p-1 arasındaki tam sayılar aralığınını kastedilir. Buradaki kasıt modulo p ile ifade edilir. Örneğin F_{23} ifadesi, 0 ile 22 arasındaki sayılar anlamına gelir. Bu küme içerisinde herhangi bir işlem yapıldığında sonuç 0 ile 22 arasında bir değer olacaktır.

Bu durumda F_p kümesinin elemanı olan her (x,y) ikilisi için yine F_p grubuna karşılık gelen bir sayı eliptik eğri üzerinde bulunabilir. Örneğin F_{23} grubu üzerinde tanımlı olan sayıları incelediğimizde a=1 ve b=0 durumu için $y^2 = x^3 + x$ denklemi elde edilir. Burada (9,5) ikilisi denklemi aşağıdaki şekilde sağlar.

$y^2 \bmod p = x^3 + x \bmod p$

$25 \bmod 23 = 729 + 9 \bmod 23$

$25 \bmod 23 = 738 \bmod 23$

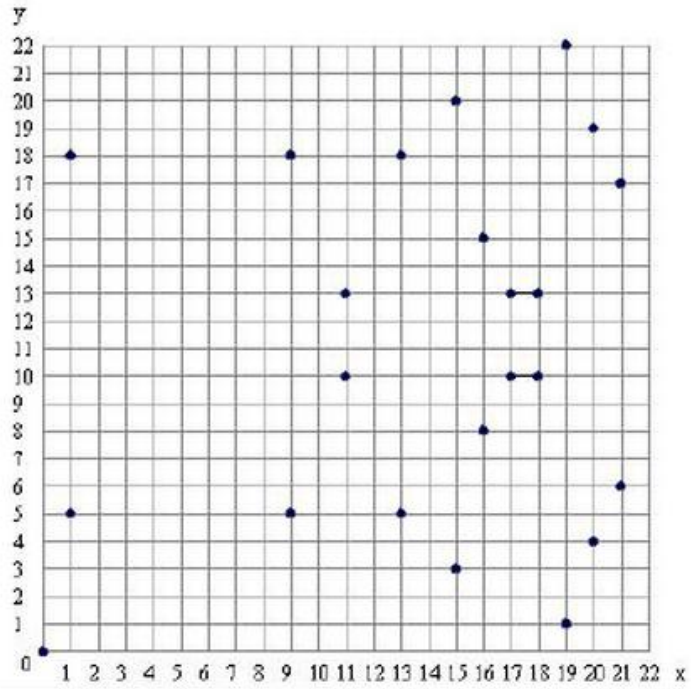
$2 = 2$

Denklemi sağlayan 23 nokta da aşağıda verilmiştir:

$(0,0)(1,5)(1,18)(9,5)(9,18)(11,10)(11,13)(13,5)(13,18)(15,3)(15,20)(16,8)(16,15)(17,10)(17,13)$

(18,10)(18,13) (19,1) (19,22) (20,4) (20,19) (21,6) (21,17) [56].

Eğer noktasal değerler koordinat düzlemine yerleştirilirse Şekil 2.10'daki grafik ortaya çıkar. Grafik her ne kadar rastgele dağılmış gibi gözükse de $y = 11.5$ doğrusuna göre simetri olduğu görülür.



Şekil 2.10. Eliptik Eğri Kullanımı [56].

F_p kümesi üzerinde tanımlı eliptik eğriler ile gerçek sayılar kümesi üzerinde tanımlı eğriler arasındaki temel fark F_p kümesi elemanlarının sonlu sayıda olmasıdır. Bu durum şifreleme için tercih edilen bir özelliktir. F_p kümesindeki tüm sayılar tam sayı olduğundan herhangi bir yuvarlama yada belirsizlik söz konusu değildir [23].

Eğer Eliptik Eğri üzerindeki $\mathrm{P(x,y)}$ noktası n kere kendini tekrar ediyorsa;

$$\mathrm{P + P + P + P \ldots . + P = nP = Q}$$

Eliptik Eğride kullanılan $\mathrm{P(x,y)}$, $\mathrm{Q(x,y)}$ değerlerini bulmak kolay olsa bile n'i bulmak çok zordur. P ve Q biliniyor olsa bile n'i bulmak çok zordur.

İşte bu ayrık logaritma problemi (ECDLP) bir sürü Eliptik Eğri(EC) ile birleşip güvenlik sistemi olan Eliptik Eğri Şifreleme Sistemi (ECC)'nin temelini oluşturur [24].

Eliptik eğri şifreleme (ECC) sonlu cisimler üzerinde eliptik eğrilerin cebirsel yapısına dayanan açık anahtarlı bir şifreleme yaklaşımıdır. Bu yaklaşım yani, eliptik eğri gruplarının sonlu alanlar üzerinde tanımlanmasının şifreleme sistemleri için temel alınması fikri ilk olarak 1985 yılında Neal Koblitz ve Victor Miller tarafından önerilmiştir [25].

Başlıca avantajı RSA ile aynı güvenlik seviyesine sahip olduğu halde ona göre daha küçük bir anahtar boyutu üreterek, saklama ve üretim maliyetini azaltmasıdır. Örneğin: 256 bitlik Eliptik Eğri (ECC) açık anahtarı, 3072 bit RSA açık anahtarıyla eşit güvenlik seviyesi sağlar. Yine 160 bitlik ECC açık anahtarı, 1535 bitlik RSA açık anahtarına eşit güvenlik seviyesi sağlar [26].

RSA şifreleme sisteminin güvenliği, büyük tamsayıların asal çarpanlarına ayrılmasındaki matematiksel zorluğuna dayanırken, Eliptik Eğri güvenilirliği yukarıda da anlatıldığı gibi RSA' e göre daha yeni bir matematiksel problem olan ve günümüzde yarı üstel zamanda hala çözülemeyen ayrık logaritma problemine (ALP) dayanmaktadır [5].

Eliptik-Eğri tabanlı protokoller için varsayılan, genel olarak bilinen bir temel noktaya göre rastgele eliptik eğri elementinin ayrık logaritmasını bulmanın imkânsız olduğudur. Bu "Eliptik Eğri Ayrık Logaritma Problemi" yâda "ECDLP" olarak isimlendirilir. Eliptik eğrinin boyutu problemin zorluğunu belirler.

Avantajları

- RSA ya göre daha kısa anahtar uzunluğu ile yüksek güvenlikli şifreleme yapılabilir.
- Anahtar uzunluğunun kısa olması bellek tüketimini azaltır.
- Anahtar uzunluğunun kısa olması işlem maliyetini azaltır.
- Sadece eliptik eğri parametreleri değiştirilerek güvenlik yeniden sağlanabilir [27].

Bilinen Saldırılar

Eliptik eğri şifreleme sistemi üstüne yapılan kriptanaliz çalışmalarından en önemlilerinden birkaçı aşağıda verilmiştir.

- Pollard'ın Rho Algoritması
- Gaudry - Hess - Smart Attack (Ghs)
- Weil Descent

Sistemi çözmek için kullanılan algoritmalardan en bilineni ve en genel amaçlı olanı Pollard'ın Rho algoritmasıdır. Tüm-üssel olarak çalışma zamanı $pr^{1/2}/2$ nokta toplamıdır.

Sabitlenmiş bir F_p alanı, Pollard'ın Rho metoduna maksimum rezistansı ile r asal olmak ve olabileceği kadar büyük bir sayı olması koşulu ile E de bir eliptik eğri seçimi yapılmaktadır. Örneğin $r \approx p$.

Kriptanalistlerin karşılaştığı zorluklar, bu eğrileri hızlı çözebilecek eliptik eğri ayrık logaritmik problemler keşfetmek olmuştur [28].

Sonlu bir A alanı eliptik eğri kriptografisinde eğer k üzerindeki tüm eliptik eğriler ayrık logaritma problemi için örnekleniyor ise örneklerin zayıf olduğu söylenmektedir. Bunlar Pollard'ın Rho metodu ile daha zor örneklerin çözümünden daha kısa sürede çözülebilmektedir [24].

Diffie-Hellman Anahtar Değişimi

Diffie-Hellman anahtar değişimi, kriptografik anahtarın değişiminde kullanılan bir yöntemdir. Bu anahtar değişimi metodu güvensiz bir ortam üzerinde karşılıklı iki tarafın ortak gizli anahtar elde edebilmelerini sağlar. Anahtar, simetrik şifreleme algoritmalarındaki simetrik anahtar olarak kullanılabilir. Bu sayede şifreleme anahtarı üzerinde anlaşan taraflar şifreli iletişimi başlatabilir [29].

Tasarım ilk kez Martin Hellman ve Whitfield Diffie tarafından 1976 yılında "New Directions in Cryptography" isimli makalelerinde yayımlanmıştır.

Sistemin Çalışma Mantığı

Sistemin çalışma mantığı basit bir matematiksel gerçeğe dayanmaktadır. Bu gerçek $g^{ab} = g^{ba}$ ifadesidir.

Aşağıda bu konu ile ilgili bir örnek verilmiştir.

İki taraf anahtar değişim yapmadan önce asal sayı olarak bir p değeri ve taban olarak bir g değeri belirlerler. Örnekte p değeri 23, g değeri ise 5 olarak seçilmiştir. Bu değerlerin üçüncü kişiler tarafından bilinmesinin bir önemi yoktur.

Taraflardan biri (Alice) gizli bir tam sayı (a) seçer. a=6

Alice bu a sayısına göre $A = g^a \ mod \ p$ işlemini yapar ve karşı tarafa (Bob) A değerini gönderir.

$A = 5^6 \ mod \ 23$ $A = 15{,}625 \ mod \ 23$ $A = 8$

Bulunan A değeri Bob'a gönderilir.

Bob gizli bir tam sayı (b) seçer. b=15. Bu b sayısına göre $B = g^b \ mod \ p$ işlemini yapar.

B=$5^{15} \ mod \ 23$ $B = 30{,}517{,}578{,}125 \ mod \ 23$ $B = 19$

Bob, Alice'e B değerini gönderir.

Alice kendisine gelen B değerini kullanarak s=$B^a \ mod \ p$ işlemini yapar ve gizli anahtarı (s) üretir.

$s_1 = B^a \ mod \ p$ $s_1 = 19^6 \ mod \ 23$ $s_1 = 47{,}045{,}881 \ mod \ 23$ s_1=2 bulunur.

Bob kendisine gelen A değerini kullanarak s=$A^b \ mod \ p$ işlemini yapar ve gizli anahtarı (s) üretir.

$s_2 = A^b \ mod \ p$ $s_2 = 8^{15} \ mod \ 23$ $s_2 = 35{,}184{,}372{,}088{,}832 \ mod \ 23$ s_2=2 bulunur.

Yukarıdaki metotta da görüldüğü üzere her iki tarafında ürettiği anahtar değerleri aynıdır. Gizli anahtar üzerinde anlaşma yapmak için taraflar arasında gidip gelen bilgi 8 ve 19 olmaktadır. Bu değerlerin herkes tarafından bilinmesinde sakınca yoktur. Yine p ve q değerlerinin de üçüncü kişiler tarafından bilinmesinde sakınca yoktur. Birbirleriyle güvensiz kanal üzerinden iletişim kurmayı amaçlayan iki taraf (Alice ve Bob) birbirlerine değerini göndermeden aynı gizli anahtarı elde edebilirler. Üçüncü kişilerin bu gizli anahtarı bulabilmeleri için Alice ve Bob'un tuttukları gizli tamsayıları (a ve b) bilmeleri gerekir. Şekil 2.11'de anahtar değişim yapısının blok şeması verilmiştir.

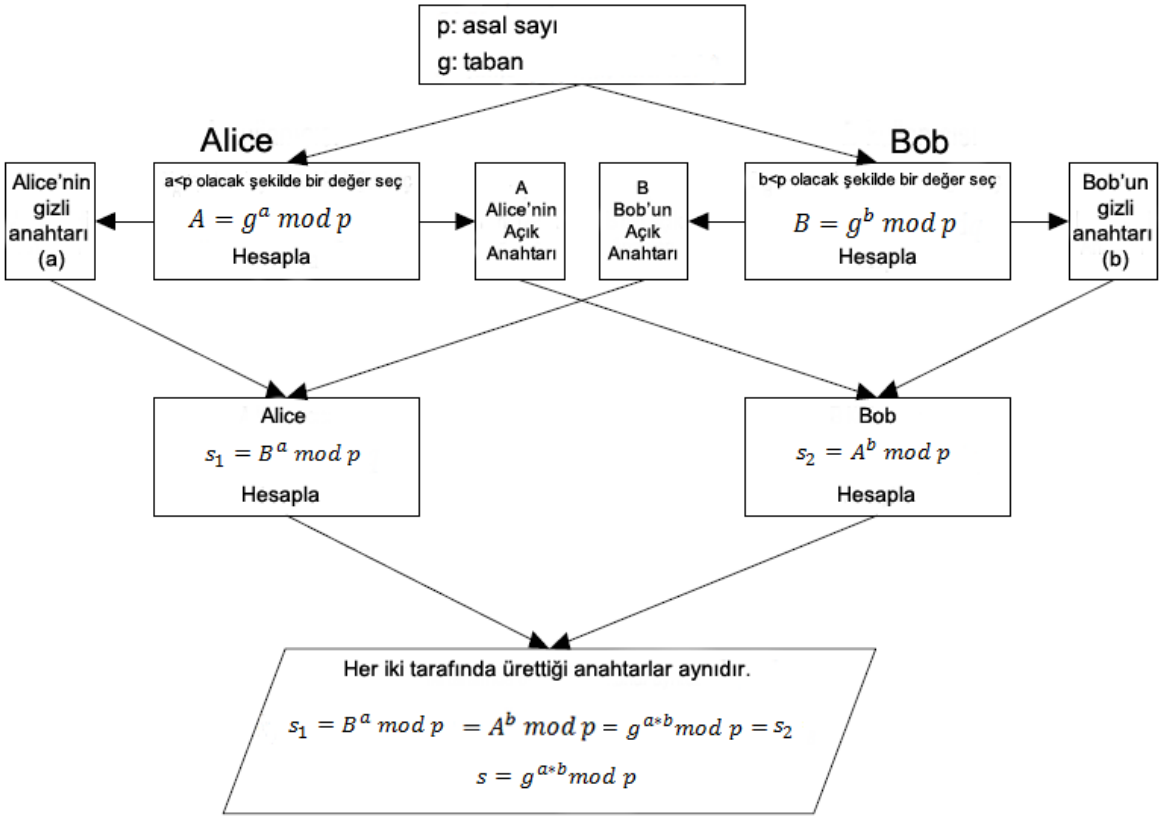
Gizli anahtar (s) bir başka ifadeyle şöyle hesaplanır.

$s = g^{a*b} mod\ p$

$s = 5^{6*15} mod\ 23$ $s = 5^{90} mod\ 23$

s=807,793,566,946,316,088,741,610,050,849,573,099,185,363,389,551,639,556,884,765,625 *mod* 23

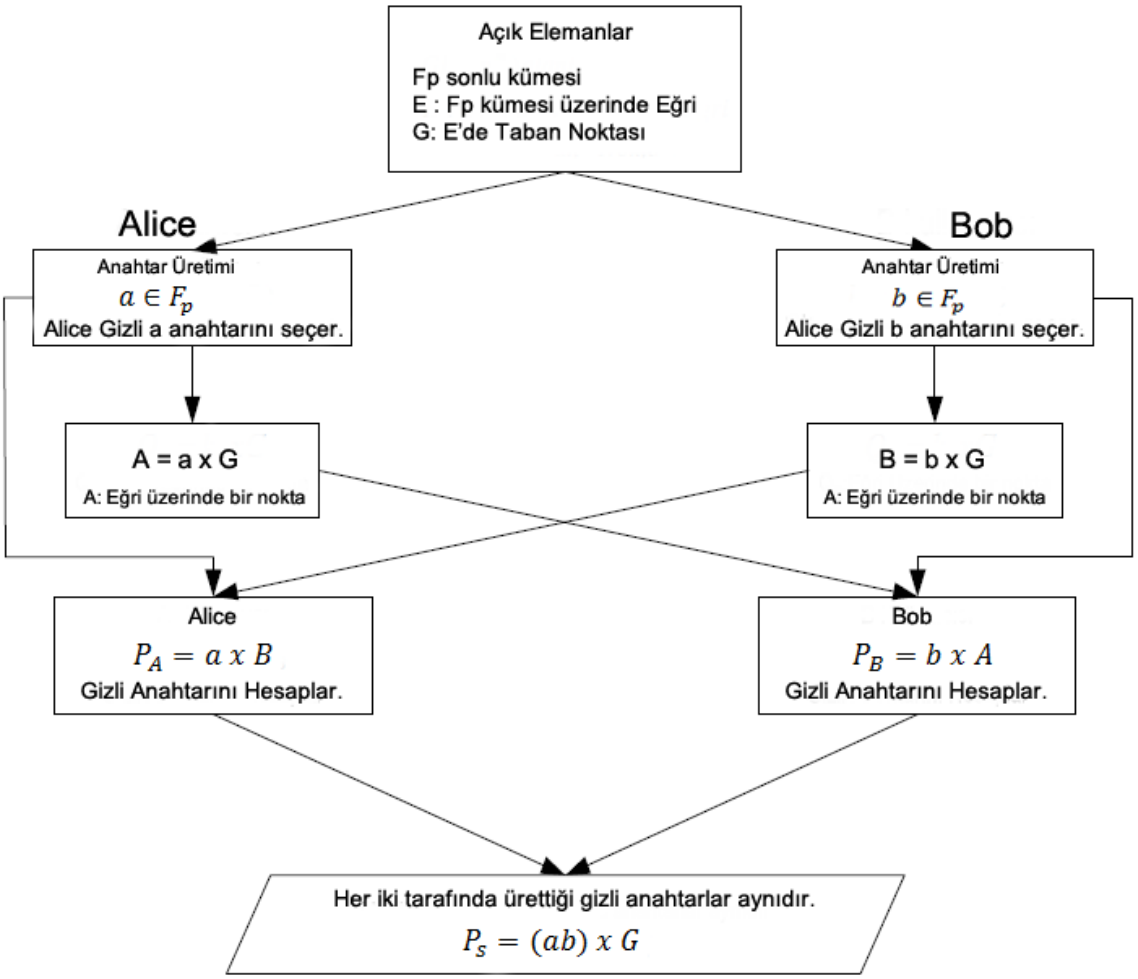
$s = 2$ bulunur.



Şekil 2.11. Diffie-Hellman Anahtar Değişim Protokolü.

Eliptik Eğri Diffie–Hellman

Yukarıda anlatılan Diffie-Helman anahtar değişim protokolü, eliptik eğriler üzerinde de uygulanabilmektedir. Bu sayede eliptik eğriler kullanılarak birbirinden bağımsız iki taraf arasında anahtar değişimi gerçekleştirilebilir. Eliptik Eğri Diffie-Hellman anahtar üretim protokolü veri akışı Şekil 2.12'deki gibidir.



Şekil 2.12. Eliptik Eğri Diffie-Hellman Anahtar Üretim Protokolü.

Bu akış sonunda oluşan $P_s = (x_s, y_s)$ noktası, E eğrisi üzerinde bir noktadır. y_s değeri, eğri denklemi ve xs değeri kullanılarak da üretilebilir. y_s' e ait ihtiyaç olunan tek bilgi, denklemden elde edilen y_s veya $-y_s$ den hangi kökün kullanılacağıdır [5].

Eliptik Eğri Diffie-Hellman protokolünün güvenliği, $P_s = (x_s, y_s)$ noktasının elde edilebilmesi için Eliptik Eğri Diffie-Hellman probleminin çözümünün zorluğuna dayanmaktadır. Verilen $G, Q \in E$ noktaları için a ve b sonlu elemanlar olmak üzere $Q = a.G$ yâda $Q = b.G$ eşitliği olsun. Eliptik Eğri Diffie-Hellman problemi bu eşitlikteki a ve b değerlerini bulmaya dayanır. Sadece G, aG, bG bilinerek P_s, a ve b hesaplanamaz. Eliptik eğri Diffie-Hellman fonksiyonları bu anlamda tek yönlü fonksiyonlardır [30].

2.2. MOBİL MESAJLAŞMA PLATFORMU

Çalışmanın bu kısmında, gerçek zamanlı mesajlaşma esnasında bildirim gönderimini sağlayan Google Cloud Messaging konusuna değinilmiş ve bir uygulama gerçekleştirilmiştir.
Google Cloud Messaging (GCM), geliştiricilerin, sunuculardan kendi android uygulamalarına veri göndermelerine yardımcı olan ücretsiz bir hizmettir. Bu hizmet kullanılarak, sunucudan android uygulamasına 4 KB'a kadar yük verisi içerebilen iletiler taşınılabilir.

Anlık maç skorlarını çeken bir uygulamanın yazıldığı varsayılsın. Bu durumda uygulamanın belirli aralıklarla, kullanılan sunucuya gidip anlık bilgileri alması gerekir. "Gol olmuş mu?" , "Maçın kaçıncı dakikası?" , "Kart gören futbolcu var mı?" vb. birçok durum için uygulamanın sunucu ile sürekli haberleşmesi gerekir. Bu durum aşağıda sıralanan dezavantajlara sahiptir.

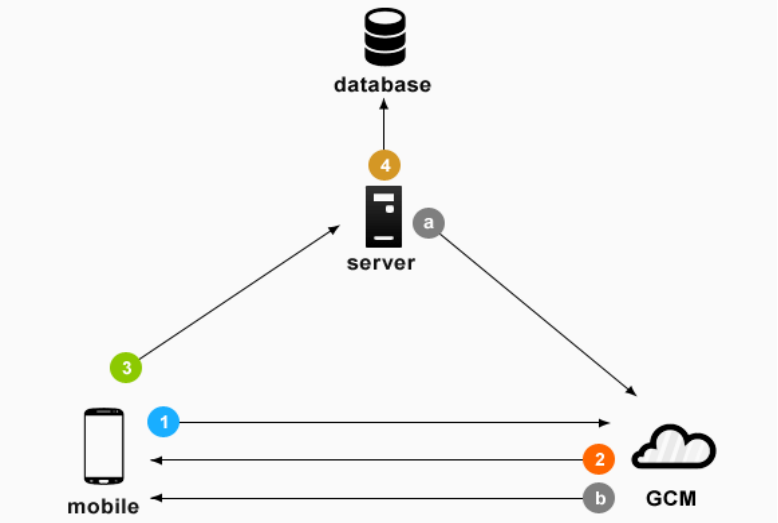
- Sunucu yoğunluğuna neden olur.
- Cihazın mobil veri aktarımı yüksek olur.
- Sürekli bir aktivite mobil cihazın daha fazla ram kullanmasına neden olur.
- Cihazın şarjı fazla kullanılacağından, bu durum bataryanın çabuk bitmesine neden olur.

Bu gibi dezvantajların önlenebilmesi için, olası değşikliklerde uygulamanın sunucuya değil bir nevi sunucunun uygulamaya çıkıp değişiklik olduğunu bildirmesi gerekir. Bu noktada Google Cloud Messaging yöntemi kullanılabilir. Bu teknoloji yardımıyla sunucu tarafından verilerde olan olası değişiklikler kullanıcının uygulamasına bildirilir. Bu sayede hem sunucu yoğunluğu azalır hem de cihazın mobil veri, ram batarya tüketimi azalmış olur.

GCM sürecinde 3 aktör görev alır.

1. Android Uygulaması
2. Google'ın GCM Sunucuları
3. Mesajlaşma Sunucu.

Bu 3 aktör arasındaki ilişki Şekil 2.13'de verilmiştir.



Şekil 2.13. GCM İlişki Yapısı.

Google Cloud Messaging teknolojisini kullanmak için öncelikle https://cloud.google.com/console adresine gidilir. Burada Google'dan GCM sunucuları ile anlaşmak için kullanılan gönderici id (sender id) ve API key bilgileri alınır. Bu bilgilerden sender id bilgisi android uygulaması içerisinde, API key bilgisi ise yazılan web servis içerisinde kullanılır.

Şekil 2.13'de verilen GCM ilişki yapısına göre;

- 1 numaralı ok ile mobil cihaz Google Cloud Messasing sunucusu ile uygulamanın (daha önce Google'dan alınan) sender id bilgisini paylaşır ve GCM sunucusuna bu bilgi ile kaydını yaptırır.
- 2 numaralı ok ile GCM sunucusu uygulamadan gelen sender id bilgisi ile uygulamayı register eder. Uygulamaya bir registration id bilgisi gönderir.
- 3 numaralı ok ile uygulama, GCM sunucusundan aldığı registration id bilgisini push notification yapacak olan sunucuya iletir.
- 4 numaralı ok ile sunucu kendisine gelen registration id bilgisini kendi veri tabanına kaydeder.

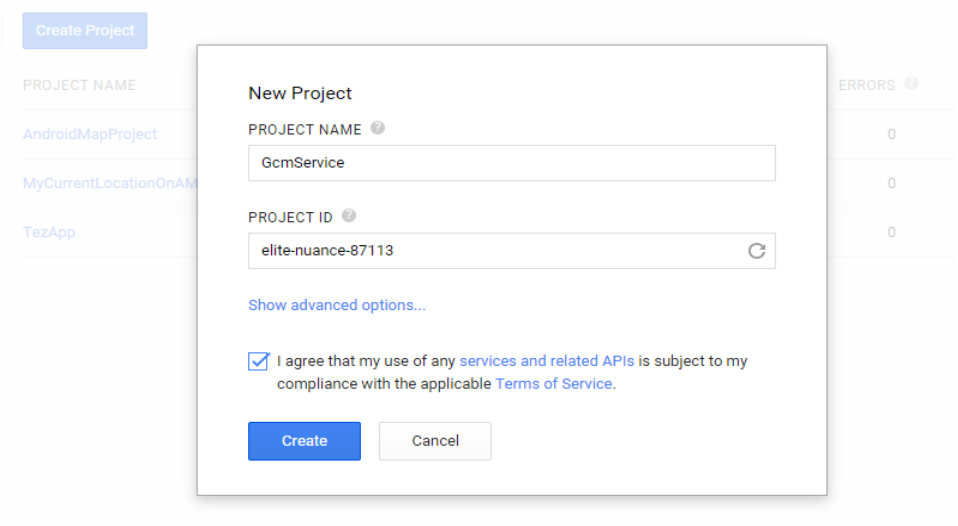
Bu işlemlerden sonra sunucu üzerinden mobil cihaza bir bilgi gönderilmek istenildiğinde sırasıyla a ve b durumları gerçekleştirilir. Sunucu cihaza bir bilgi göndermek istediğinde, GCM Api key bilgisini kullanarak ilgili mobil cihazın

registration id'si ile birlikte gönderilmek istenen bilgi GCM sunucusuna iletir (a durumu). GCM sunucusu kendisine gelen registration id'yi kullanarak o registration id'sine sahip olan mobil cihaza bilgiyi iletir (b durumu).

Google Cloud Messaging teknolojisinin çalışma mantığı bu şekildedir. Aşağıda bu teknolojinin nasıl kullanıldığına değinilmiştir.

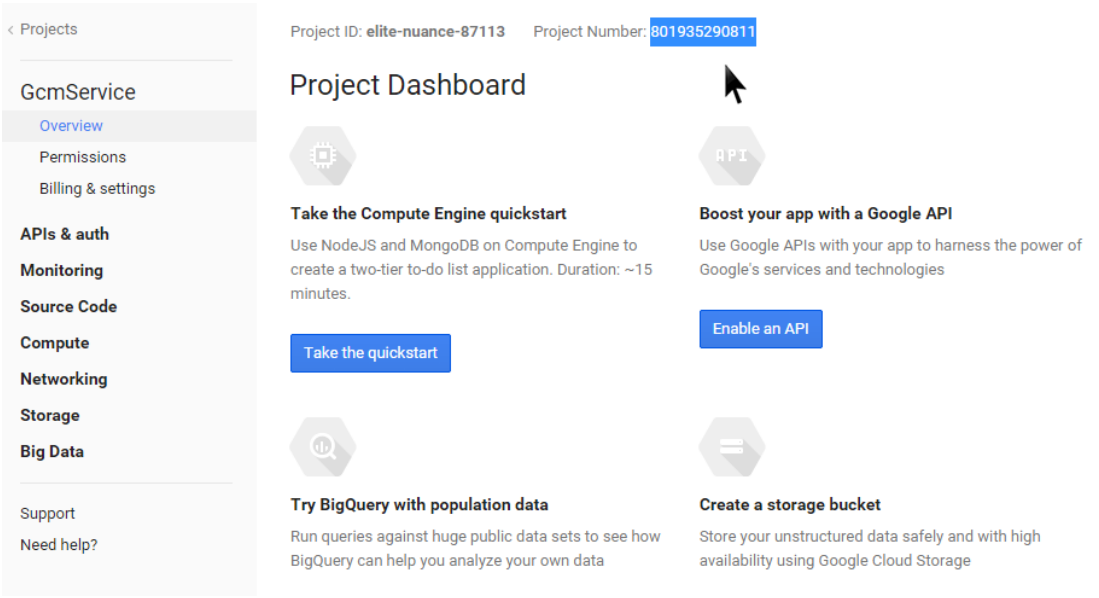
2.2.1. Gcm Kullanarak Android Push Notification İşlemi

Öncelikle Şekil 2.14'de verildiği gibi Google API Console sayfasını açıp "Create Project" butonuna basarak yeni bir proje oluşturulur.



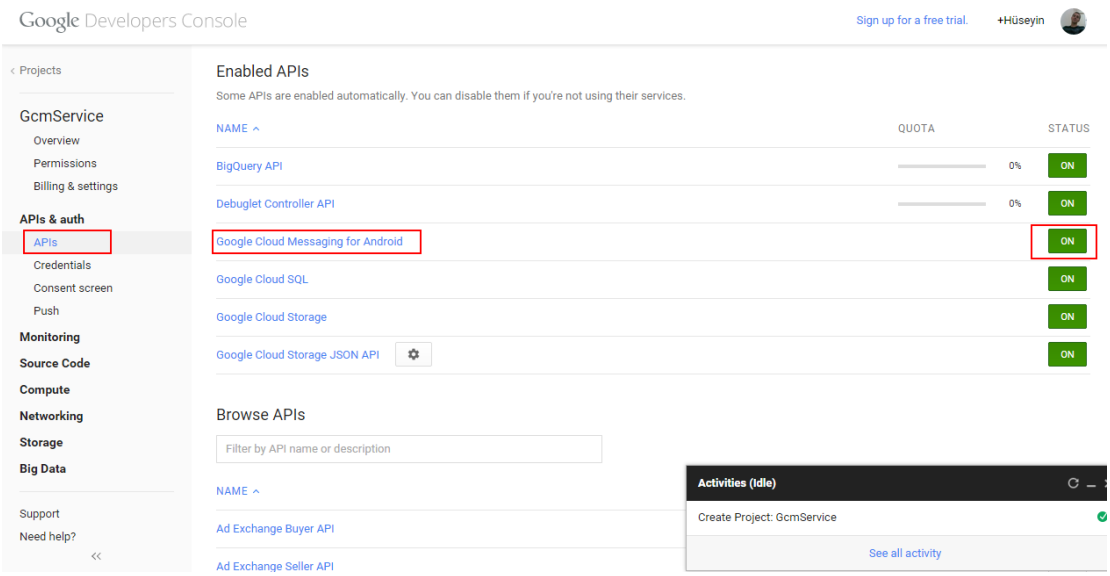
Şekil 2.14. Google Proje Oluşturma.

Şekil 2.15'de görüldüğü üzere android projenin içerisinde kullanılacak olan sender id bilgisi (project number) bir kenara not edilir.



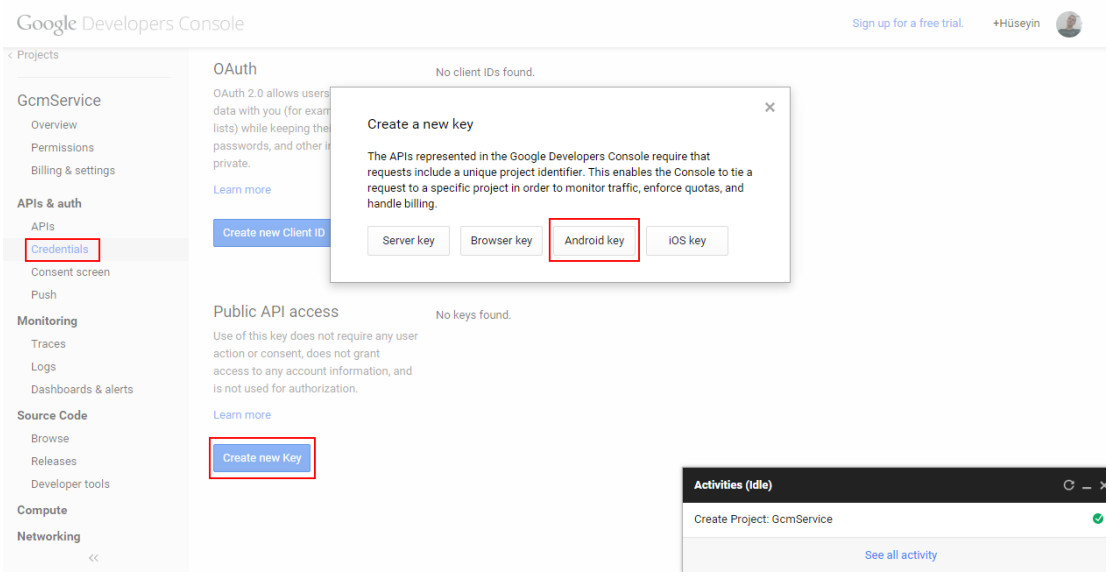
Şekil 2.15. Google Proje Oluşturma 2.

Şekil 2.16'da işlem adımları verilen GCM yapısı üzerinde API & auth altındaki API sekmesine gidilip Google Cloud Messaging for Android servisi aktif hale getirilir.



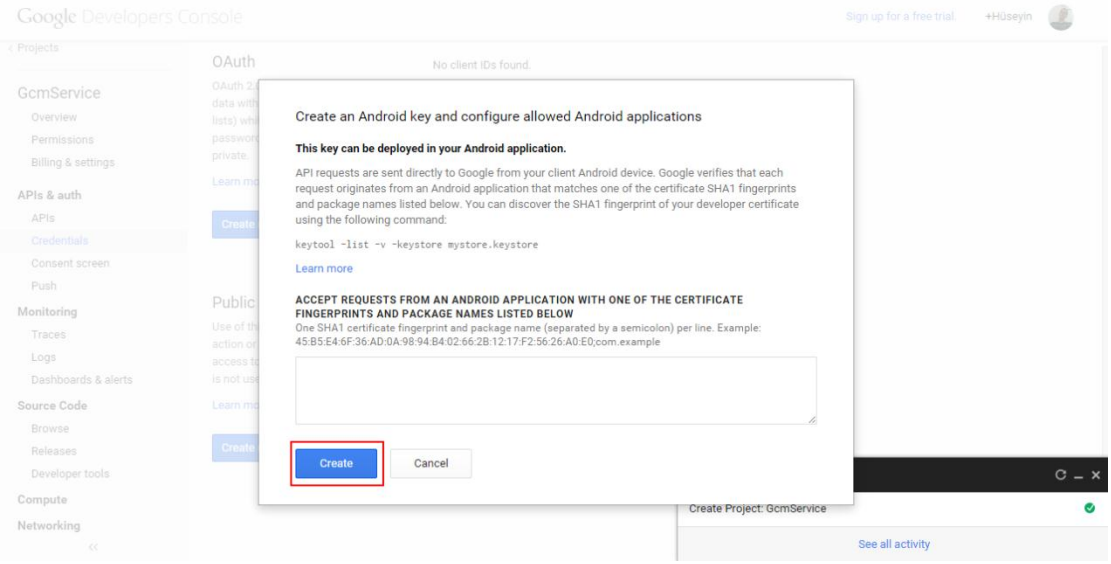
Şekil 2.16. Google API Aktif Hale Getirme.

Ardından şekil 2.17'de görüldüğü üzere, API & auth altında Credentials sekmesine girilip, Create New Key butonuna tıklanır ve gelen pencerede Android Key seçeneği seçilir.



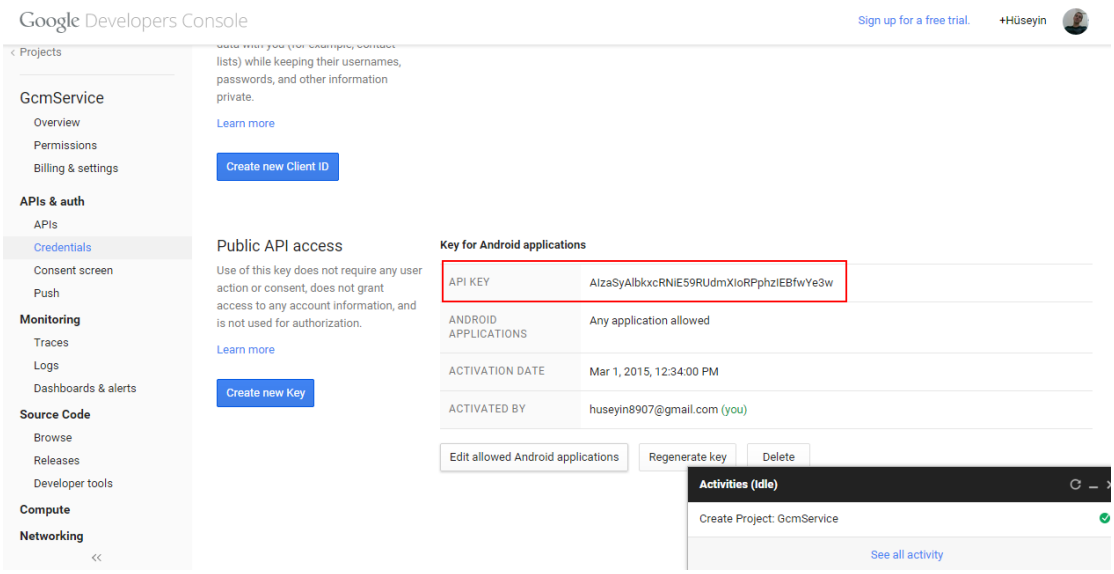
Şekil 2.17. Android Key Elde Etme.

Şekil 2.18'de görüldüğü üzere açılan pencerede yeniden Create butonuna basılır.



Şekil 2.18. Android Key Elde Etme 2.

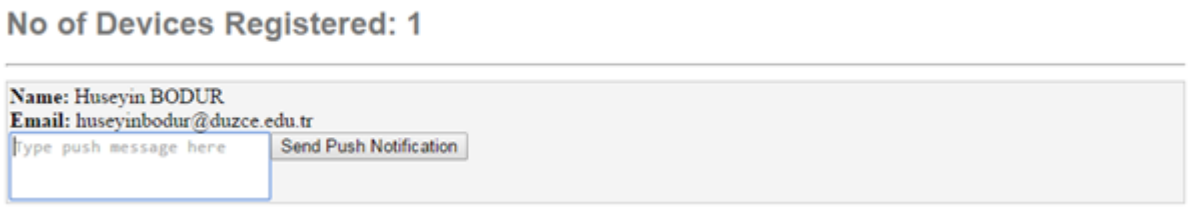
Şekil 2.19'da görüldüğü üzere, API key'i kullanıma hazır hale geldi. Bu key değeri, php web servisi içerisinde kullanılacağından dolayı bir kenara not edilir.



Şekil 2.19. Android API Key.

Google Cloud Messaging uygulamasının php kod yapısı, Ek-1'de CD ortamında sunulmuştur.

Index.php dosyası ile ekranda kayıtlı olan kullanıcılar (mobil cihaz ile gcm_users dosyasına kayıtlı olan) listelenebilir. Şekil 2.20'de görüldüğü üzere dosya üzerinden kayıtlı kullanıcılardan istenilen birine, push notificacation yöntemiyle mesaj gönderilebilir.

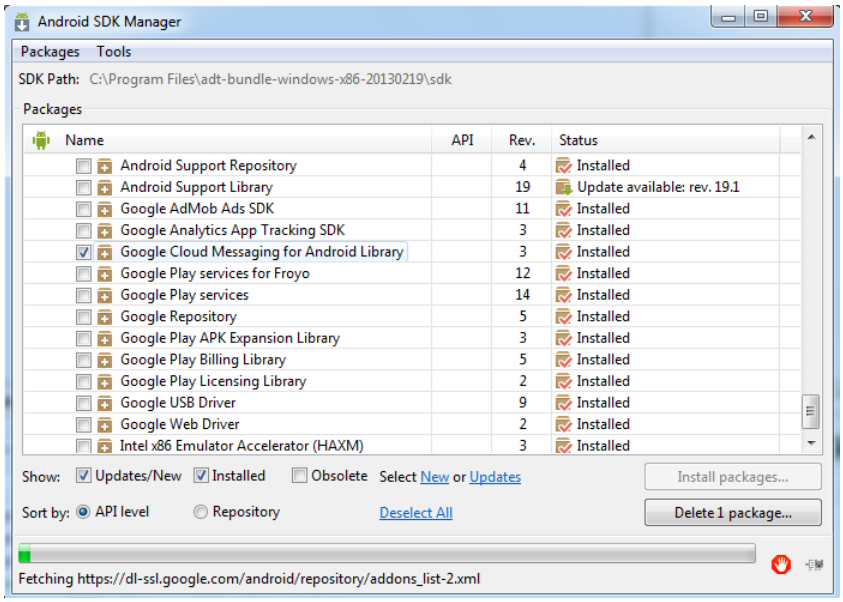


Şekil 2.20. Notification Gönderme.

2.2.2. Google Cloud Messaging Android Uygulaması

Google Cloud Messaging uygulamasının Android kısmı aşağıda verilmiştir.

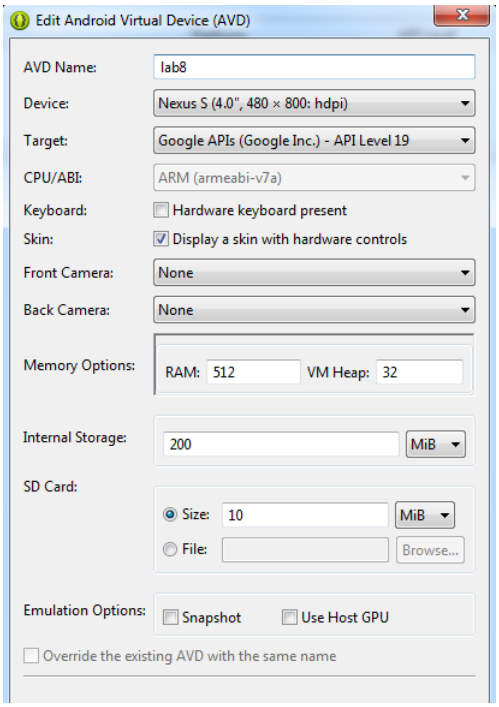
Öncelikle yeni bir android projesi oluşturulur. Projenin SDK Manager kısmına gidilir, Şekil 2.21'de görüldüğü üzere açılan pencerede Extras sekmesinin altında Google Cloud Messaging Library kurulur.



Şekil 2.21. Google Cloud Messaging Kütüphanesini Yükleme.

Proje yapısının libs klasörü altına gcm.jar dosyası eklenir. Bu jar dosyası registration id değeri üretilirken kullanılır.

Şekil 2.22'de görüldüğü üzere, bir sonraki adımda uygulamanın test edileceği sanal cihazın ayarları yapılır. GCM işleminin başarıyla gerçekleştirebilmesi için sanal cihazda hedef olarak Google APIs seçilmesi gerekir.



Şekil 2.22. Google API'sine Uygun Emülatör Oluşturma.

Projenin Androidmanifest.xml dosyası içerisine girip gerekli izinleri tanımlanır.

INTERNET : Uygulama içerisinde internet kullanımına izin verir.

ACCESS_NETWORK_STATE : Network durumuna erişmek için gerekli izindir.

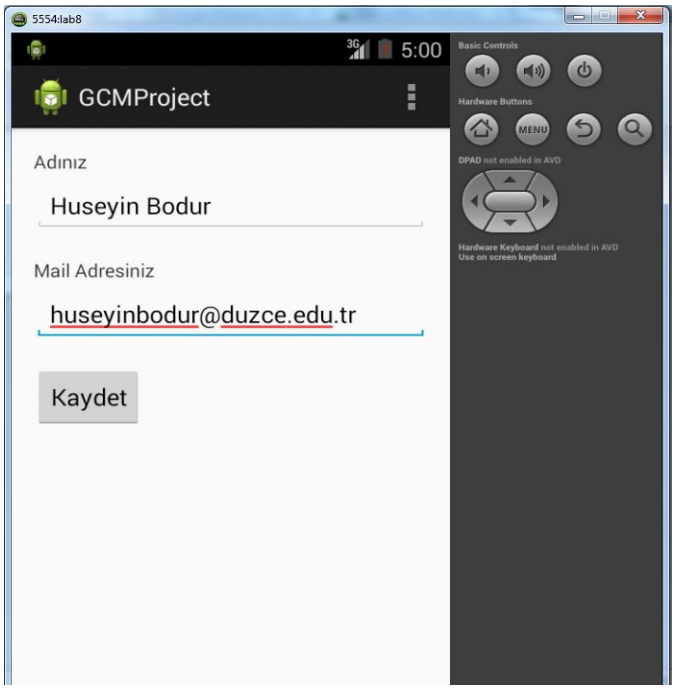
GET_ACCOUNTS : GCM için gerekli olan Google Accounts iznidir.

WAKE_LOCK : Uygulamanın uyku vaziyette iken uyanması gerekirse bu izin gereklidir.

VIBRATE : Notification geldiğinde titreşim özelliğine izin verir.

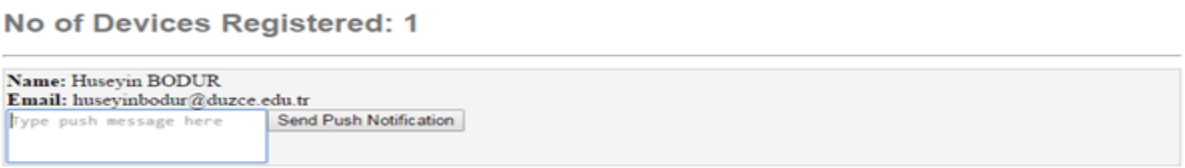
Google Cloud Messaging uygulamasının android kod yapısına, ek olarak verilen media ortamından ulaşabilirsiniz.

Şekil 2.23'de görüldüğü üzere, kullanıcının adı, mail bilgisi ve GCM registration id bilgisi alınıp sisteme kayıt edildikten sonra proje çalıştırılıp örnek bir push notification işlemi gerçekleştirilebilir.



Şekil 2.23. Kullanıcı Bilgilerini Sisteme Yükleme.

Şekil 2.24'de görüldüğü üzere web sayfasında veritabanındaki kayıtlar listelenir.

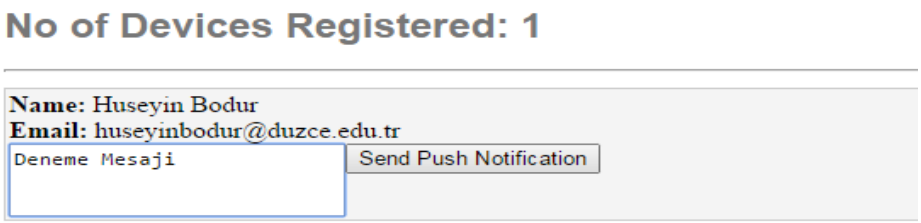


Şekil 2.24. Kullanıcı Bilgilerinin Web Yapısında Görüntülenmesi.

Şekil 2.25'de görüldüğü üzere kayıtlara veritabanı içerisinden de ulaşılabilir.

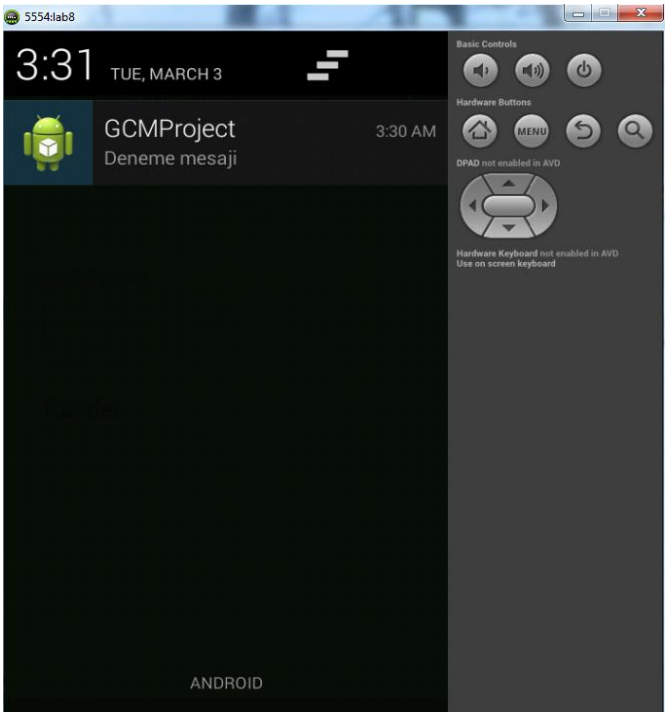
Şekil 2.25. Kullanıcı Bilgilerinin Veritabanında Görüntülenmesi.

Şekil 2.26'da görüldüğü üzere örnek bir push notification işlemi gerçekleştirilir.



Şekil 2.26. Örnek Bir Push Notification İşleminin Gerçekleştirilmesi.

Şekil 2.27'de görüldüğü üzere "Send Push Notification" butonuna basıldığında ekrana push notification mesajının düştüğü görülür.



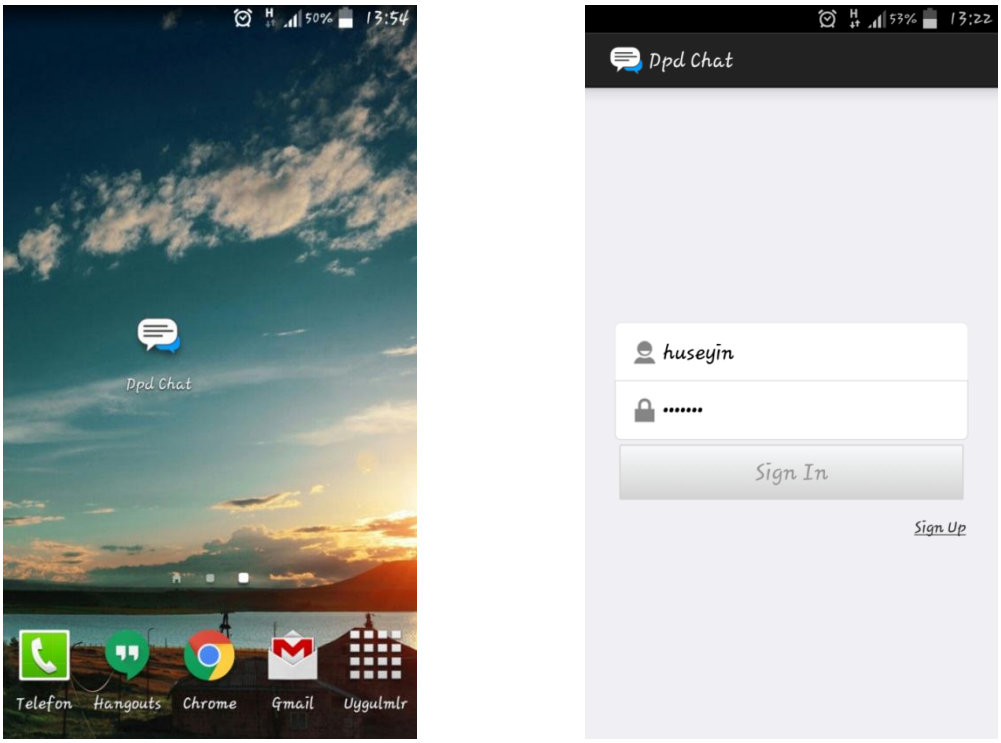
Şekil 2.27. Notification'un Uygulama Ekranında Görüntülenmesi.

Google Cloud Messaging uygulamasının android kod yapısı, Ek-2'de CD ortamında sunulmuştur.

2.3. ANDROİD ŞİFRELEME UYGULAMASI

2.3.1. Kullanıcı Giriş

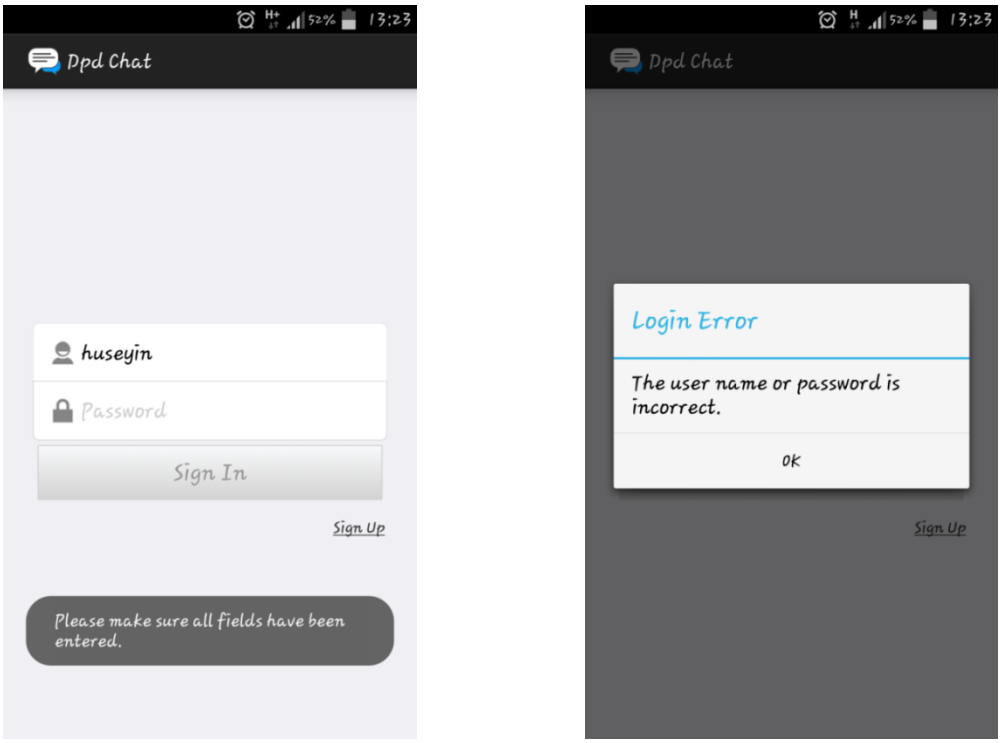
Şekil 2.28'de görüleceği üzere, giriş ekranında kullanıcıya, kullanıcı adı ve şifresi sorulur. Kullanıcı sisteme kayıtlı olduğu kullanıcı adı ve şifre ile giriş yapar.



Şekil 2.28. Kullanıcı Giriş Ekranı.

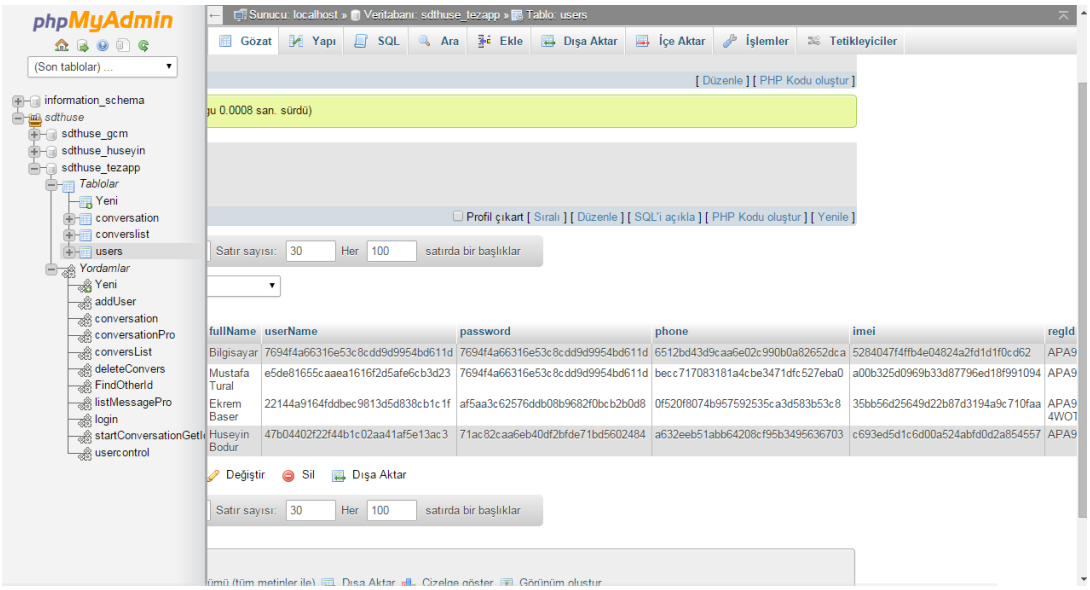
Burada önemli olan nokta uygulamanın sadece kullanıcı adı ve şifreyi değil aynı zamanda uygulamanın kullanıldığı cihazın Imei numarasını da kontrol etmesidir. Bu sayede kullanıcı sistemde sadece kayıt olduğu cihaz ile oturum açabilir. Kullanıcı, uygulamanın yüklü olduğu başka bir cihazda kendi kullanıcı adı ve şifre bilgilerini doğru girse bile oturum açamaz.

Şekil 2.29'da giriş sayfası üzerinde kontrol işlemleri gösterilmetedir.



Şekil 2.29. Giriş Ekranı Kontrollleri.

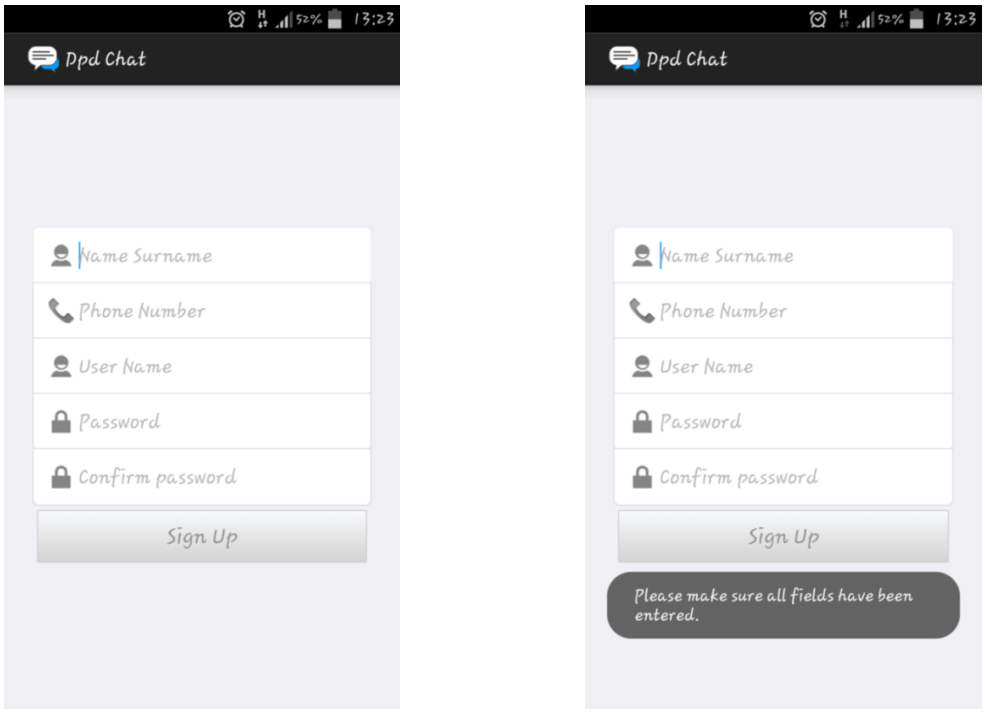
Şekil 2.30 'da görüldüğü gibi kullanıcı uygulamaya giriş yapmak istediğinde internet üzerindeki Mysql veritabanına kullanıcı bilgileri şifreli olarak gönderilir. Burada bilgiler doğrulama işlemine tabi tutulur ve sonuç yeniden uygulamaya gönderilir.



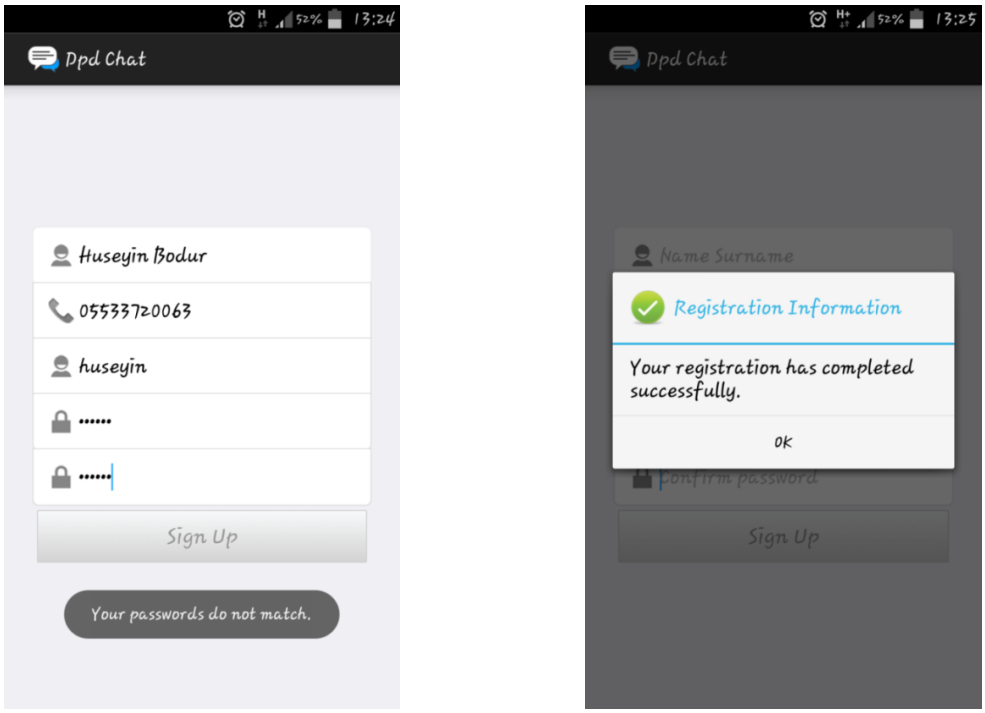
Şekil 2.30. Kullanıcı Bilgilerinin Veritabanı Görüntüsü.

2.3.2. Kullanıcı Kayıt

Şekil 2.31 ve Şekil 2.32'de kullanıcı kayıt sayfası ve kontrolleri verilmiştir.

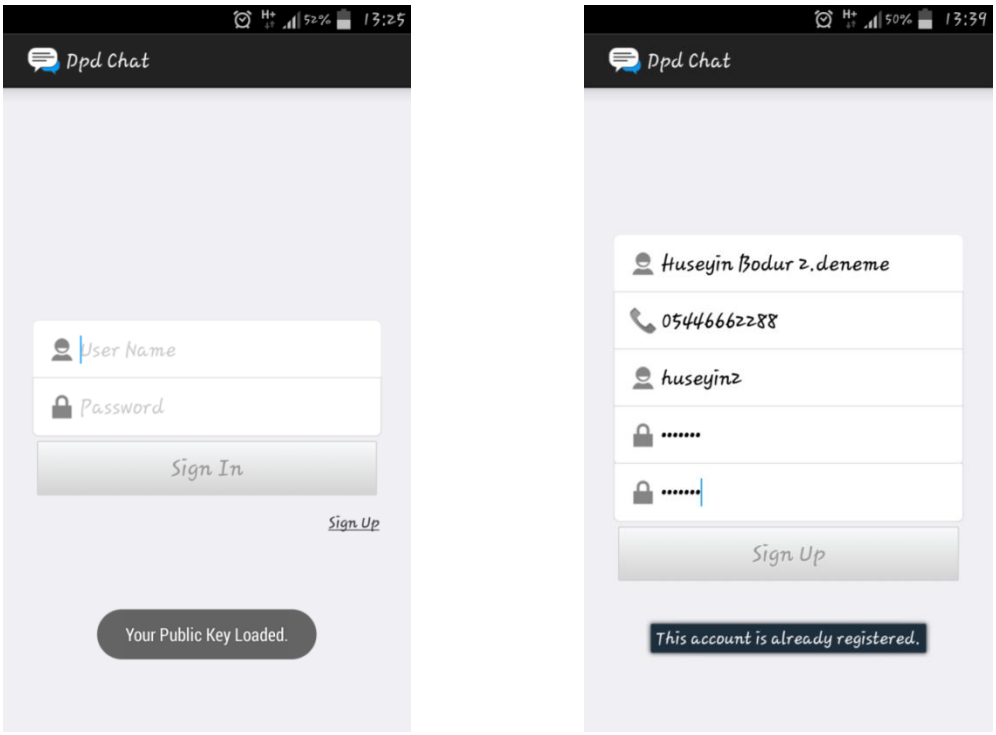


Şekil 2.31. Kullanıcı Kayıt Ekranı ve Kontrolleri.



Şekil 2.32. Kullanıcı Kayıt Ekranı Kontrolleri ve Başarılı Kayıt İşlemi.

Şekil 2.33'de kullanıcı anahtarlarının mobil ve web ortamlarına aktarımı görüntülenmektedir.



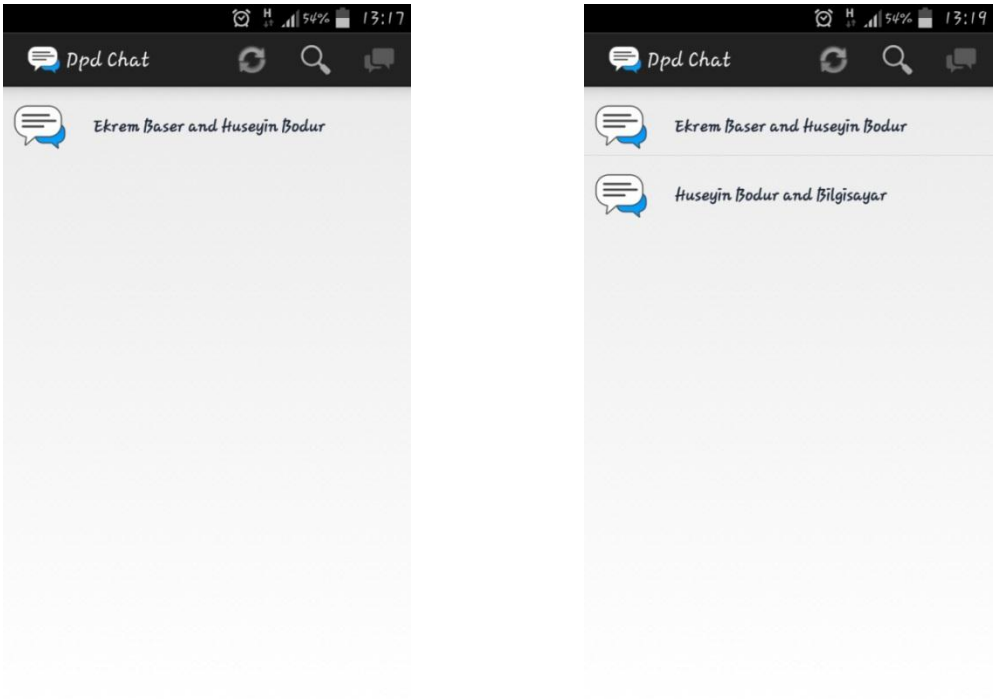
Şekil 2.33. Kullanıcı Key Bilgilerinin Mobil ve Web Sistemlerine Yüklenmesi.

Uygulamayı ilk defa telefonuna yükleyen kişinin bir defaya mahsus olmak üzere sisteme kaydını yaptırması gerekir. Burada kullanıcıdan adı soyadı, telefon numarası, şifresi gibi bilgiler istenir. Arka planda ise telefonun Imei numarası ve kullanıcının Push Notification işlemini gerçekleştirebilmesi için Google Cloud Messaging teknolojisi ile üretilen registration id bilgisi alınır. Kullanıcının adı, soyadı ve registration id bilgilerinin dışındaki diğer tüm veriler şifrelenerek gönderilir.

Şekil 2.30 'a bakıldığında kullanıcının isim bilgisi ve adı soyadı ve registration id bilgisi haricindeki diğer tüm bilgilerin şifrelenerek tutulduğu görülür. Buradaki şifreleme metodu MD5 şifrelemedir. Sistem kullanıcının ikinci defa kayıt yapmasına izin vermez. Kullanıcı sisteme kayıt olur olmaz, kendi cihaz sistemi üzerinde açık ve gizli anahtarı oluşturulur. Aynı zamanda oluşturulan açık anahtarı internet üzerinde bulunan anahtar kütüphanesine yüklenir. Bir kullanıcı mesaj atmak istediği kişiye mesajını göndermeden önce, mesaj göndereceği kişinin açık anahtarını anahtar kütüphanesinden indirir, ardından göndereceği metni bu anahtar değeri ile şifreleyerek gönderir.

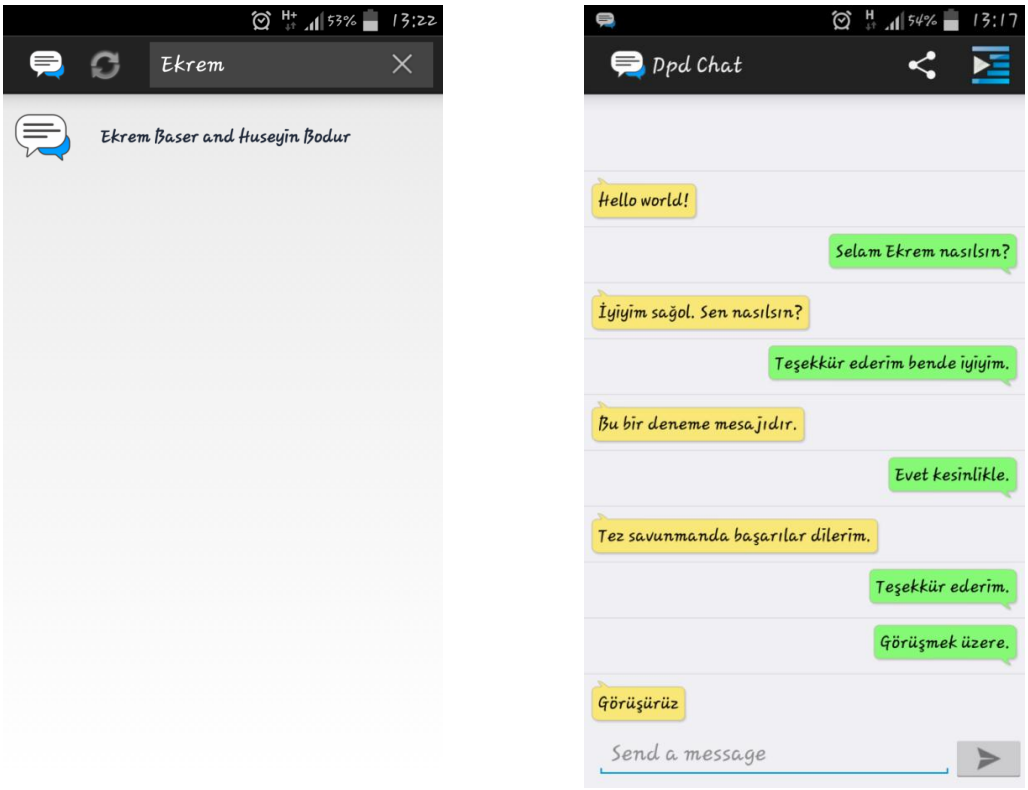
2.3.3. Mesaj Listeleme

Kullanıcı başarılı giriş yaptığında listeleme ekranına gönderilir. Burada kullanıcının önceki konuşmalarının bulunduğu liste mevcuttur. Şekil 2.34'de görüldüğü üzere, kullanıcı bu listeden istediği bir konuşmayı seçerek eski konuşmalarını görebilir ve yeni mesaj gönderebilir.



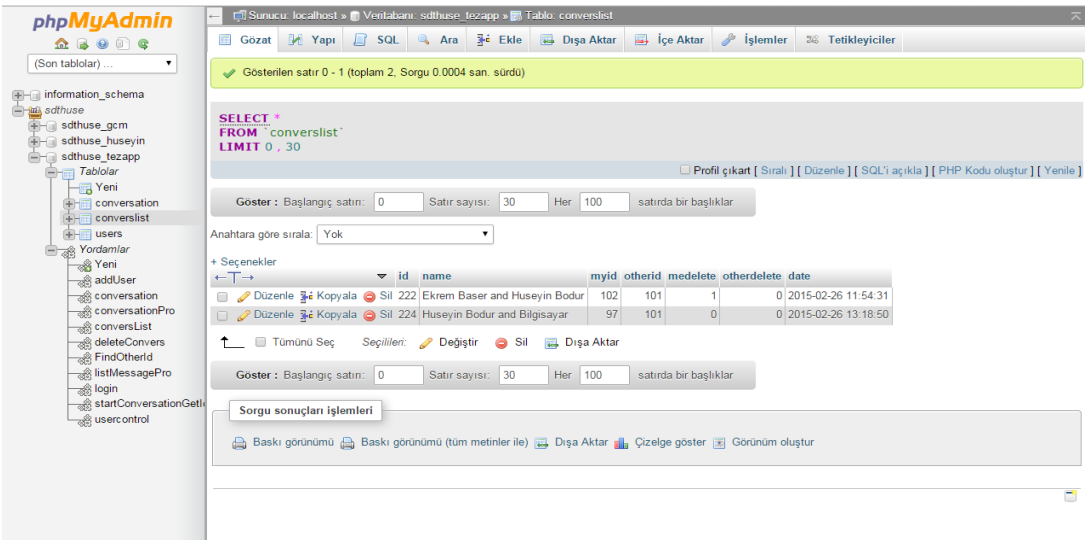
Şekil 2.34. Mevcut Mesajlaşmaların Uygulama Arayüzünde Listelenmesi.

Şekil 2.35'deki mesajlaşma ekranında ayrıca listenin olası güncellenmemesi durumunda Refresh butonu, liste içerisinde arama yapabilmesini sağlayacak Search butonu ve yeni bir konuşma başlatabilmesini sağlayan ve telefonunun rehberini listeleyen sayfaya yönlendirme butonu bulunur.



Şekil 2.35. Mevcut Mesajlaşmaların Ekranda Listelenmesi.

Konuşma listeleri Şekil 2.36 'da görüldüğü üzere internet üzerinde Mysql veritabanında converslist tablosunda tutulur.

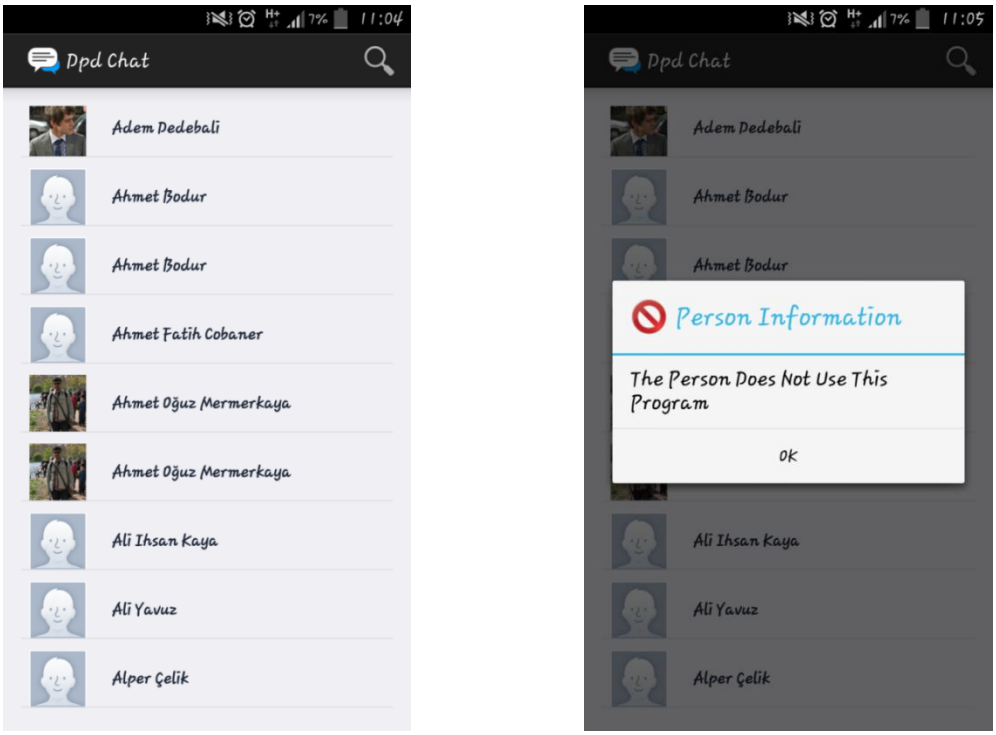


Şekil 2.36. Mesaj Listelerinin Veritabanı Görüntüsü.

Tablo sütunlarına bakıldığında, içerisinde “and” bağlacının kullanıldığı mesajlaşma adını tutan name sütunu, mesajlaşmayı gerçekleştiren kişilerin id bilgilerini tutan myid ve otherid sütunu, mesajı silen kişinin yeniden konuşma listesini görmemesi için kullanılan medelete, otherdelete sütunları ve konuşmanın başlatıldığı tarih bilgisini tutan date sütununun olduğu görülür.

2.3.4. Telefon Rehberi

Şekil 2.37’de görüldüğü üzere, kullanıcı yeni bir mesajlaşma işlemi başlatmak istediğinde bu rehber üzerinden, mesajlaşmak istediği kişiyi seçer ve mesajlaşma ekranına yönlendirilir.

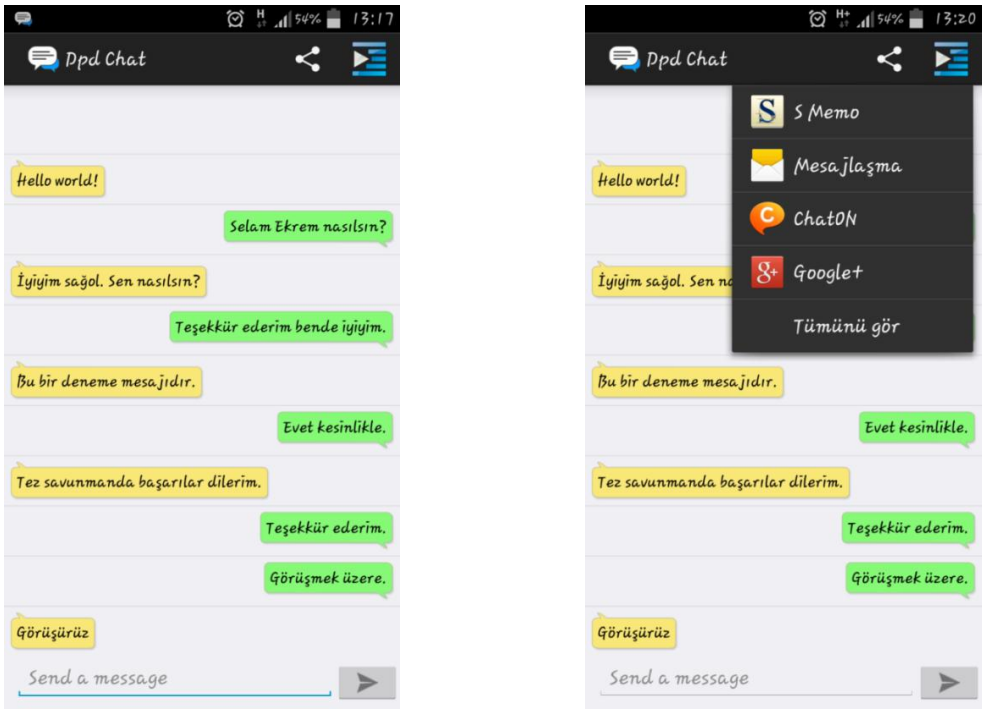


Şekil 2.37. Yeni Bir Mesajlaşma Başlatma Ekranı.

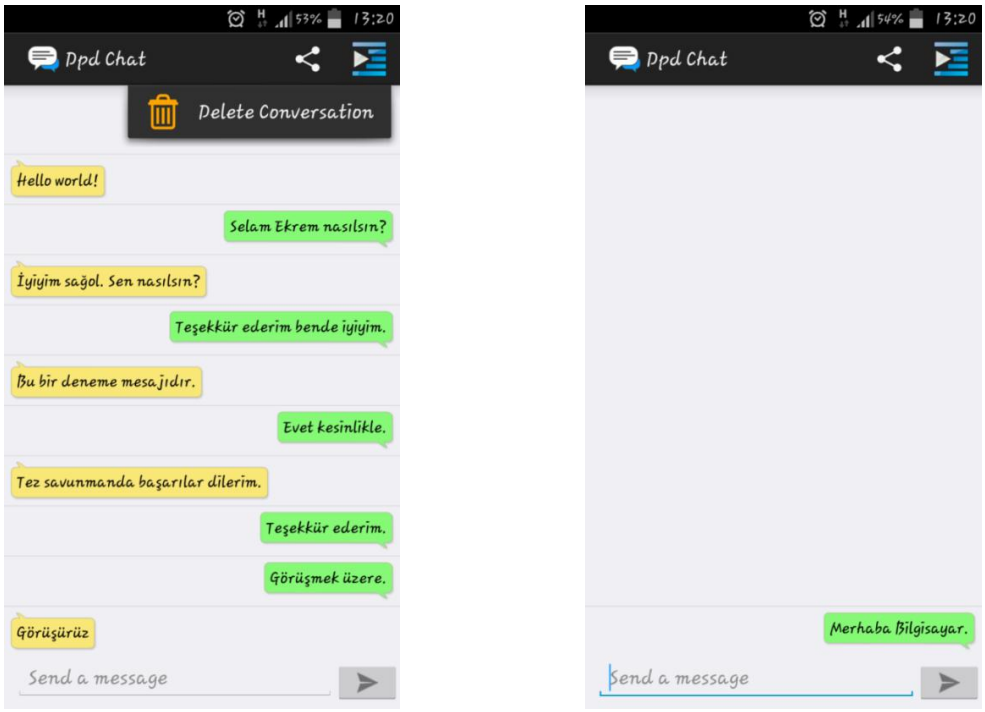
Eğer mesajlaşmak istediği kişi uygulamayı kullanmıyorsa, ekranda uyarı mesajı belirir.

Şekil 2.38’de mesajlaşma, Şekil 2.39’da ise mevcut mesajlaşmayı silme işlemleri görüntülenmektedir.

2.3.5. Mesajlaşma



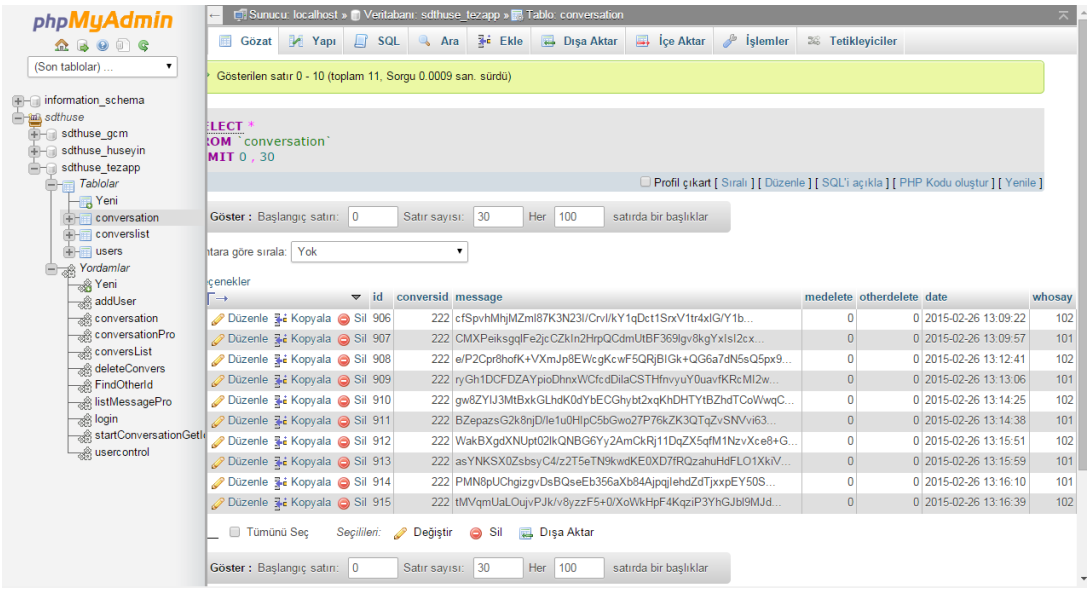
Şekil 2.38. Mesajlaşma Ekranı.



Şekil 2.39. Mevcut Mesajlaşmayı Silme Ekranı.

Kullanıcının mesaj alıp gönderdiği ekrandır. Kullanıcı var olan mesajlaşma listesinden birini seçerek yâda yeni bir mesajlaşma işlemi başlatarak bu ekrana yönlendirilebilir. Mesajları gönderme işleminde, mesaj öncelikle gönderilecek kişinin Public Key bilgisi ile şifrelenir. Ardından Php web servisleri aracılığıyla hem Mysql veritabanına kaydedilir, hem de alıcıya iletilir. Alıcının mobil cihazına şifreli mesaj düşer düşmez, uygulama otomatik olarak kullanıcının Private Key bilgisi ile şifreli metni çözer. Anlamlı metni elde eder ve listeler. Mesaj, ayrıca hem mesajı gönderen hem de alan kişinin cihazının veri tabanında kendi Public Key'leri ile şifrelenerek tutulur. Bu mesajlaşma verileri arttıkça tüm verilerin her seferinde Mysql veritabanından çekilmemesini sağlayarak hız ve performansı arttırır.

Şekil 2.40'da görüleceği gibi mesajlaşma işlemi sırasında göndericiden alıcıya giden veri şifrelenip, internet üzerinde Mysql veritabanında bulunan conversation tablosuna gönderilir.



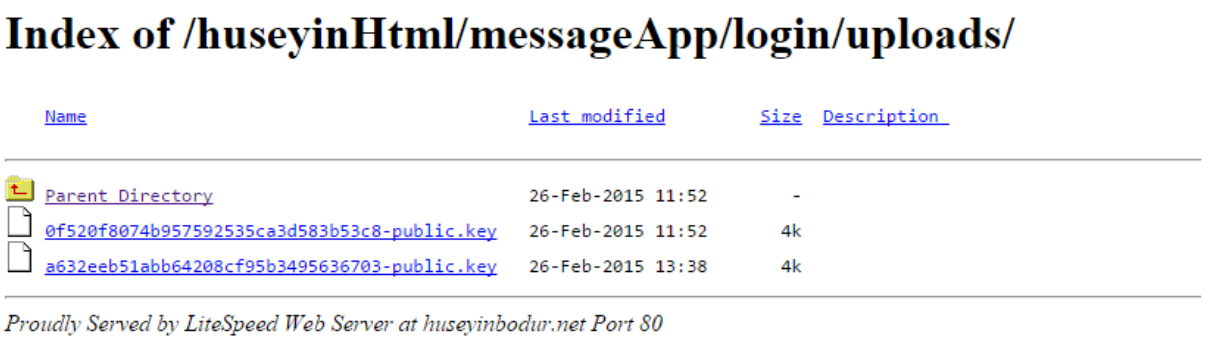
Şekil 2.40. Mesajlaşmaların Veritabanı Görüntüsü.

Conversation tablosuna bakıldığında, içerisinde mesajlarının hangi kullanıcıya ait olduğunun ayırt etdilmesini sağlayan conversid sütununun bulunduğu görülür. Ayrıca şifreli verilerin tutulduğu message sütunu, mevcut mesajlar silinmek istenildiğinde hangi mesajların silindiğinin ayırt edilmesini sağlayan medelete ve otherdelete sütunları, mesajların gönderim tarihini tutan date sütunu, mesajları gönderen kullanıcının ayrımını yapmayı sağlayan ve kullanıcının id bilgisini tutan whosay sütunu görülür.

Veri tabanına kayıt işlemini yapan php dosyası eşzamanlı olarak şifreli metni karşı kullanıcının mobil cihazına, kullanıcının registration id bilgisini kullanarak gönderir. Push notification teknolojisi ile gönderilen şifreli veri karşı kullanıcının cihazına ulaşır ulaşmaz çözümlenir ve anlamlı halde listelenir.

Kullanıcı mesajlaşmak istediği birini seçip mesaj gönderme ekranına yönlendirildiğinde, mesaj göndereceği kişinin açık anahtarı kendi cihaz sisteminde mevcut değil ise uygulama, internet üzerinde bulunan anahtar kütüphanesine giderek ilgili kişinin açık anahtarını cihaz sistemine dâhil eder. Bu işlem bir defaya mahsus gerçekleştirilir.

Şekil 2.41'de kullanıcıların açık anahtarlarını yükledikleri ve indirdikleri anahtar kütüphanesini görüyoruz.



Şekil 2.41. Açık Anahtar Kütüphanesi.

Anahtar kütüphanesinde her kullanıcının anahtar ismi şifrelenerek tutulur.

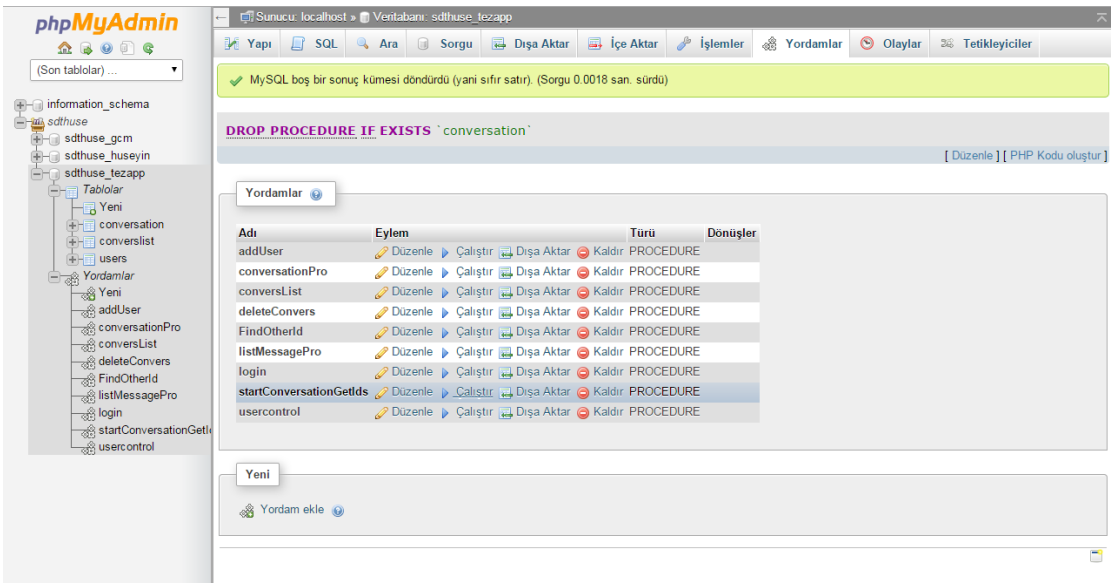
Mesajlaşma ekranında değinmemiz gereken bir diğer konu Delete Conservation işlemidir. Kullanıcı bir konuşmayı silmek istediğinde konuşmanın içinde sağ üst açılır liste kutusuna tıklayıp, Delete Conversation işlemini yapması gerekir. Bu durumda mesajlar kullanıcının cihazının veri tabanından silinir. Eşzamanlı olarak internet üzerinde Mysql veritabanı üzerinde bulunan converslist ve conversation tablolarında tutulan ilgili mesajların medelete, otherdelete sütunları güncellenir.

Güncelleme işleminin ardından, arka planda çalışan prosedure hem converslist hem de conversation tablosunu kontrol eder ve içerisindeki kayıtlardan hem medelete sütunu hem de otherdelete sütunu "1" olan tüm satırları siler. İki sütunun da "1" değerinde

olması, hem göndericinin hem de alıcının mevcut konuşmayı silmiş olduğu anlamına gelir.

Uygulamanın web kısmında, Mysql veritabanı üzerinde, yukarıda da açıklamalarını yapmış olduğumuz, 3 adet tablo bulunur.

Bunlar; users, converslist ve conversation tablolarıdır. Bu tablolar üzerinde select, insert, update ve delete gibi tüm işlemler, Php web servis aracılığıyla yine Mysql veritbanı üzerinde oluşturulmuş Stored Procedure'ler kullanılarak gerçekleştirilir. Şekil 2.42'de uygulama içerisinde kullanılan Stored Procedure'ler verilmiştir.



Şekil 2.42. Uygulamanın Web Kısmında Kullanılan Storage Procedure'ler.

Uygulamanın hem java hem de php web servis yapısının kaynak kodları Ek-3'de CD ortamında sunulmuştur.

3. BULGULAR VE TARTIŞMA

3.1. RSA

RSA şifrelemede anahtar boyutu seçimi büyük bir önem taşır. Anahtar boyutu arttıkça sistemin güvenlik seviyesi, karmaşıklığı ve şifreli metinlerin direnci artar. Bu avantajlar şifreli metinlerin çözülmesini, şifre kırılma ihtimallerini zorlaştırır. Fakat bu avantajların yanı sıra, şifreleme anahtarının oluşturulma süresi, metinlerin şifrelenme süresi ve mobil cihazın ram tüketimi de artar. Bu dezavantajlar uygulamanın etkin bir şekilde kullanımını etkileyecek unsurlardır. Bu nedenle anahtar boyutlarının avantaj ve dezavantajlarının belirlenip, en uygun olan anahtar boyutu tercih edilmelidir.

RSA şifreleme anahtarının seçimindeki bir diğer önemli unsur, şifreleme anahtar değerine karşılık tek seferde en fazla kaç karakter mesajın şifrelenebileceğidir. Bu durum anahtar boyutu ile doğru orantılıdır ve anahtar boyutu arttıkça artış gösterir. Çizelge 3.1'de tek seferlik iletimde bir anahtar boyutu ile en fazla kaç karakterlik şifreleme işlemi yapılabilineceğini gösteren tablo verilmiştir.

Çizelge 3.1. RSA Algoritması Anahtar Boyutları.

Boyut	Mesaj Uzunluğu	Şifre Uzunluğu
256 bit	0-32 karakter	44 karakter
512 bit	0-64 karakter	88 karakter
1024 bit	0-128 karakter	172 karakter
2048 bit	0-256 karakter	344 karakter
3072 bit	0-384 karakter	512 karakter
4096 bit	0-512 karakter	684 karakter
8192 bit	0-1024 karakter	1368 karakter

Örneğin 512 bitlik RSA şifreleme anahtarı seçildiğinde en fazla 64 karaktere kadar mesajı tek seferde şifreleyip, gönderilebilir. Bu durumun nedeni her karakterin 8 bit değere karşılık gelmesidir.

Eğer tek seferde 64 karakterden daha fazla boyutta bir mesaj gönderilecekse;

Bu durumda çözüm yollarından biri, daha büyük boyutta bir şifreleme anahtarı kullanmaktır ki bu durumun avantaj ve dezavantajlarına yukarıda değinilmiştir.

Bir diğer çözüm yolu ise, şifrelenecek mesaj üzerinde sıkıştırma yâda bloklara ayırma metotları uygulamaktır. İki metottan sadece birisi uygulanabileceği gibi her iki metotta aynı anda uygulanabilir. Sıkıştırma metodu için önerilen, metnin öncelikle sıkıştırılması daha sonra şifreleme işlemine tabi tutulmasıdır. Bunun nedeni metin içindeki tekrarlı durumların sıkıştırma metodu ile tespit edilebilmesinden kaynaklanmaktadır.

Bloklara ayırma yani metni parçalara ayırma durumunda ise anahtar boyutu yine büyük bir önem taşır. Metnin kaç bloğa ayrılacağı doğrudan anahtar boyutuna göre belirlenir. Aşağıda 512 bitlik bir anahtar boyutu seçilmesi durumunda metnin kaç bloğa ayrılması gerektiğini gösteren basit bir pseudo kod verilmiştir.

512 bit anahtar boyutu için;

$$If\ (\ mesaj\_boyutu > 64\ and\ mesaj\_boyutu\ mod\ 64\ = 0)$$

$$blok\_sayisi = mesaj\_boyutu/\ 64;$$

$$else\ If\ (mesaj\_boyutu > 64\ and\ mesaj\_boyutu\ mod\ 64\ != 0)$$

$$blok\_sayisi = (\ (tamsayi) mesaj\_boyutu/64\ ) + 1;$$

(3.1.1)

Çizelge 3.2'de farklı anahtar boyutlarının mobil cihazlar üzerinde ortalama oluşturulma süreleri, belirlenen bir metnin farklı anahtar boyutlarında şifrelenme süreleri, şifreli metinlerinin alıcı tarafa iletim süreleri ve anahtarların kaydedildikleri dosya boyutları görülmektedir.

Çizelge 3.2. Farklı Anahtar Uzunluklarının Oluşturulma Süreleri.

Anahtar Boyutu	Anahtar Oluşturma Süresi	Şifreleme Süresi	İletim Süresi	Boyutu
256 bit	20 ms	7 ms	588 ms	182 bayt
512 bit	125 ms	8 ms	632 ms	214 bayt
1024 bit	525 ms	12 ms	531 ms	278 bayt
2048 bit	2938 ms	16 ms	565 ms	406 bayt
3072 bit	7704 ms	20 ms	675 ms	534 bayt
4096 bit	16236 ms	54 ms	1194 ms	662 bayt
8192 bit	156571 ms	54 ms	3095 ms	1174 bayt

3.2 ELİPTİK EĞRİ

Eliptik Eğrilerin denklemi : $y^2 = x^3 + ax + b \; mod \; p$ dir.

Burada y, x, a ve b değerleri büyük sayısal değerlerdir. Ayrıca denklem değerlerinin $4a^3 + 27b^2 \neq 0$ koşuluna duyarlı olması gerekmektedir.

Eğri Kripto Sisteminin Parametreleri

- p: Büyük bir asal sayıdır. F_p üzerindeki tüm işlemler mod p alınarak yapılır.
- a ve b: Eğri için tanımlamış iki katsayı değeridir.
- G: Üretici yâda ana noktadır. Eğrinin başlangıç noktasıdır. Ya büyük bir formu şeklinde yâda iki ayrı g_x ve g_y gibi sayısal değer olarak verilir.
- n: Eğri üzerindeki farklı noktaların sayısıdır.
- h: Eğrinin kofaktörüdür.

Örneğin secp160k1 eğrisi için yukarıdaki parametrelerin rakamsal değerleri şöyledir. Anahtar değerleri ve parametreler Daniel R.L.Brown 'un "Recommended Elliptic Curve Domain Parameters" isimli çalışmasından elde edilmiştir. Brown çalışmasında Eliptik Eğri ile oluşturulabilecek şifreleme yapıları için farklı anahtar boyutlarını ve bu boyutlarda anahtar üretmek için gerekli olan parametreleri listelemiştir [31].

```
secp160k1.p = "FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFAC73";
secp160k1.a = "0000000000000000000000000000000000000000";
secp160k1.b = "0000000000000000000000000000000000000007";
secp160k1.G = "023B4C382CE37AA192A4019E763036F4F5DD4D7EBB";
secp160k1.n = "0100000000000000000001B8FA16DFAB9ACA16B6B3";
secp160k1.h = 1;
```

Android mobil platform ortamında Eliptik Eğri ile şifreleme örneği yapabilmek için Spongycastle isimli jar dosyasının indirilip uygulama içerisine kütüphane (libs) olarak eklenmesi gerekir.

Uygulamamız içerisinde kullanılabilecek eğri isimleri aşağıda verilmiştir. Bu isimler Spongycastle tarafından desteklenen eğri isimleridir.

Uygulamanın Desteklediği Eğri İsimleri

sect283k1, secp128r1, secp128r2, sec160r1, sect163r2, sect163r1, sect193r2, secp192r1, sect163k1, secp160r2, sect571r1, sect193r1, sect233r1, secp256k1, secp256r1,

sect131r1, secp224k1, sect131r2, sect239k1, sect113r1, sect283r1, scp521r1, secp224r1, sect233k1, secp384r1, sect409k1, sect571k1, sect113r2, secp160k1, secp192k1, sect409r1, secp112r1, secp112r2

Uygulama geliştirirken doğru eğri tipini seçmek önemlidir. Çünkü anahtar boyutu seçilen eğri tipine göre belirlenir. Anahtar değeri şifreli metnin gücü ile doğru orantılıdır. Anahtar boyutu arttıkça şifreli metnin gücü de artar.

Anahtar Çifti Üretmek

Gizli anahtar üretmek için rastgele bir sayı seçilir: d_A öyle ki 0<d_A <n olmalıdır.

Açık anahtar üretmek için özel anahtar d_A değeri G üretici değeri ile noktasal çarpma işlemine sokulur. $Q_A = d_A . G$

Açık ve gizli anahtar değerleri RSA yönteminde sayısal değerler iken, eliptik eğri yönteminde gizli anahtar değeri sayısal bir değer, açık anahtar değeri ise eğri üzerinde bir noktadır.

Eliptik eğri sisteminde örnek olarak seçilen anahtar uzunluk değerleri, o anahtar değerlerine karşılık gelen eğri isimleri ve anahtar değerlerine karşılık gelen RSA anahtar uzunluk değeri Çizelge 3.3'de listelenmiştir [32].

Çizelge 3.3. ECC Algoritması Anahtar Boyutları.

ECC Anahtar Uzunluğu	Anahtar Boyutu	RSA Anahtar Uzunluğu	ECC Adı
160 bit	40 karakter	1024 bit	secp160k1
192 bit	48 karakter	1536 bit	secp192k1
224 bit	56 karakter	2048 bit	secp224k1
256 bit	64 karakter	3072 bit	secp256k1
384 bit	96 karakter	7680 bit	secp384r1
521 bit		15360 bit	secp521r1

Çizelge 3.3'de de görüldüğü gibi 160 bitlik anahtar uzunluğuna sahip bir eliptik eğri yapısı, 1024 bitlik RSA anahtar boyutuna denk güçtedir. Aynı şekilde 15360 bit gibi çok güçlü bir RSA anahtar boyutuna karşılık 521 bitlik eliptik eğri anahtarı kullanmak yeterlidir. Anahtar boyutu arttıkça şifreleme, şifre çözme, güç ve kaynak tüketimi gibi nedenlerin arttığını belirtecek olursak, 15360 bitlik RSA anahtarı yerine 521 bitlik

eliptik eğri anahtarı kullanmak sistemin hız ve performansına büyük bir avantaj sağlayacaktır. Küçük anahtar boyutları ile karmaşık ve güçlü bir şifreleme sistemi oluşturmak için RSA algoritması yerine Eliptik Eğri anahtarını kullanan bir şifreleme sistemi tasarlamak çok daha doğru bir karar olacaktır.

Çizelge 3.4'de Eliptik eğri sistemindeki farklı anahtar boyutlarının oluşturulma, şifreleme ve iletim süreleri listelenmiştir.

Çizelge 3.4. Farklı Anahtar Uzunluklarının Oluşturulma Süreleri.

Anahtar Boyutu	**Anahtar Oluşturma Süresi**	**Şifreleme Süresi**	**İletim Süresi**	**Boyutu**
160 bit	985 ms	4 ms	253 ms	295 bayt
192 bit	139 ms	8 ms	216 ms	335 bayt
224 bit	105 ms	10 ms	255 ms	379 bayt
256 bit	113 ms	9 ms	287 ms	423 bayt
384 bit	363 ms	10 ms	294 ms	595 bayt
521 bit	402 ms	10 ms	344 ms	721 bayt

Özellikle şifreleme sürelerinin çok küçük değere denk gelmesi, şifreleme işlemi için kullanılan anahtar boyutuyla doğru orantılıdır.

Eliptik Eğri - RSA karşılaştırması için güvenlik seviyeleri birbirlerine denk olan anahtar boyutları ve o anahtar boyutlarıyla yapılan şifreleme süreleri karşılaştırıldığında aşağıdaki tablo elde edilir.

Eliptik eğri sisteminde, RSA şifreleme sistemine karşılık düşük anahtar boyutlarıyla yüksek güvenlik seviyelerinin sağlanması büyük bir avantajdır. Bu avantajın yanı sıra Çizelge 3.4'de görüldüğü üzere Eliptik eğri sistemi düşük şifreleme sürelerine de sahiptir.

Şifreleme sürelerinin RSA sistemine göre daha düşük olmasının nedeni, sistemin daha düşük anahtar boyutlarına sahip olmasıdır. Eliptik eğri sisteminin anahtar uzunlukları kullanılarak elde edilecek şifreleme sürelerinin RSA sistemine göre avantajı sistemin performans artışı olarak Çizelge 3.5'de görülebilir.

Çizelge 3.5. Eliptik Eğri ve RSA Sistemlerinin Karşılaştırması.

ECC Anahtar Uzunluğu	Şifreleme Süresi	RSA Anahtar Uzunluğu	Şifreleme Süresi
160 bit	4 ms	1024 bit	7 ms
192 bit	8 ms	1536 bit	8 ms
224 bit	10 ms	2048 bit	12 ms
256 bit	9 ms	3072 bit	16 ms
384 bit	10 ms	7680 bit	20 ms
521 bit	10 ms	15360 bit	54 ms

Çizelge 3.6'da Eliptik Eğri sistemi tercih edildiği takdirde RSA sistemine göre farklı anahtar boyutlardaki performans artışı görüntülenmektedir.

Çizelge 3.6. Eliptik Eğri Sisteminin Avantajı.

ECC Anahtar Uzunluğu	Performans Artışı
160 bit	+ %87.5
192 bit	%0
224 bit	+%60
256 bit	+%89
384 bit	+%100
521 bit	+%270

Eliptik Eğri sisteminin RSA sistemi ile karşılaştırıldığında bir diğer avantajı ise RSA şifreleme sisteminde anahtar boyutunun şifreleyebileceği maksimum bir mesaj değerinin olmasıdır. Bu konuyu biraz daha açmamız gerekirse, 512 bitlik anahtar boyutu kullandığımız RSA şifreleme algoritmasıyla 0-64 karakter yâda 2048 bitlik anahtar boyutu kullandığımız RSA şifreleme algoritmasıyla 0-256 karakter uzunluğunda mesajlar tek seferde şifrelenip gönderilebilir. Daha uzun boyuttaki mesajlar için şifreleme işlemi öncesi bloklara ayırma yâda sıkıştırma gibi işlemler uygulanmalıdır. Şifreleme sistemi içerisine giren bir mesajın çıktı olarak elde edileceği karakter uzunluğu bellidir. Bu uzunluk 512 bitlik bir anahtar boyutu için 88 karakter, 2040 bitlik bir anahtar boyutu için 344 karakterdir.

Eliptik eğri sisteminde ise farklı anahtar boyutları için eğri çeşitleri mevcuttur ve şifreleme işlemi RSA sisteminde olduğu gibi anahtar kullanılarak yapılır. Fakat bu sistemde şifrelenecek mesajın uzunluğu için bir kısıtlama yoktur. Mesaj uzunluğu ne olursa olsun şifreleme yapısından geçer ve şifrelenir. RSA sisteminde farklı anahtar boyutları için farklı ve sabit şifreli metin boyutları oluşurken, eliptik eğri sisteminde mesajın şifrelenmiş hali, şifrelenecek mesajın uzunluğuna göre farklılık gösterir.

Çizelge 3.7'de görüldüğü gibi şifrelenecek mesaj 8 karakter arttıkça, mesajın şifreli hali de 12 karakterlik bir artış göstermektedir.

Çizelge 3.7. Şifreli / Şifresiz Mesaj Uzunlukları.

Şifrelenecek Mesajın Uzunluğu	Şifreli Mesajın Uzunluğu
0-7 karakter	12 karakter
8-15 karakter	24 karakter
16-23 karakter	32 karakter
24-31 karakter	44 karakter
32-39 karakter	56 karakter
40-47 karakter	64 karakter
48-55 karakter	76 karakter
56-63 karakter	88 karakter
64-71 karakter	96 karakter
72-79 karakter	108 karakter

Ayrıca şifrelenecek mesajın uzunluğu her 32 karakterde bir 12 karakterlik değil 8 karakterlik bir şifreli mesaj uzunluğuna neden olmaktadır. Bu durum doğrudan şifreleme işleminde kullanılan anahtarla ilgilidir.

Eğer şifrelenmek istenilen mesajdan elde edilecek şifre değerinin sabit boyutta olmasını istiyorsak RSA şifreleme algoritmasını kullanmak daha doğru bir seçim olacaktır. Aksi durumlarda Eliptik Eğri sistemini terci etmek çok daha uygundur.

RSA şifreleme anahtarlarının oluşturulma, şifreleme ve mesajın iletim sürelerine bakacak olursak yüksek boyutlardaki RSA anahtarlarının oldukça hızlı olduğunu fark ederiz. Fakat yine de 3072 bit ve üzeri anahtar boyutlarının mobil tabanlı uygulamalarda kullanılmamasını öneririz. Çünkü yapılan çalışmalarda bu anahtar boyutu ve üzeri için anahtar oluşturma, şifreleme ve iletme sürelerinin oldukça fazla olduğu ve bu boyutlarda işlem yapan mobil cihaz üzerinde yavaşlamaların meydana geldiği fark edilmiştir.

Bu noktada ayrıca akıllı cihazın işlemci yapısı da büyük bir önem taşımaktadır. Mobil uygulamayı destekleyen android sürümüne karşın, mobil cihazın donanımsal yapısı uygulama cihaza yüklense dahi kullanım sıkıntısı yaratabilir. Çünkü bu donanımsal yapı uygulamanın arkaplanda yaptığı karmaşık şifreleme ve şifre çözme işlemleri için uygun olmayabilir. Uygulamamız Android 4.0 ve üzeri sürümler için uygundur ve yazılımsal olarak bu sürüm ve üzeri sürümlere yüklenebilir.

Metnin şifreli halinin sabit boyutlarda değil karmaşık boyutlarda olması isteniyorsa Eliptik eğri sistemini kullanmak bir diğer avantajdır. Bu durumun dezavantajı şifreli metni eline geçirebilen ve şifreleme algoritması olarak eliptik eğri sisteminin kullanıldığını bilen birinin orijinal mesaj boyutunun karakter aralığını tahmin edebilmesidir. Fakat yine de bu bilgilerden şifreli metni çözme ihtimalinin imkânsız olduğu düşünülmektedir. Bunun nedeni eliptik eğri sisteminin dayanağı olan ayrık logaritma probleminin zorluğudur.

Bulgular, android işletim sistemine sahip, 1.6 GHz dört çekirdek + 1.2 GHz dört çekirdek olmak üzere 8 çekirdek CPU ve 8 GB bellek kapasitesine sahip mobil cihaz ile 1.2 GHz dört çekirdek ve 16 GB bellek kapasiteli mobil cihaz üzerinde incelenerek elde edilmiştir.

4. SONUÇLAR VE ÖNERİLER

Eğer bir uygulama aracılığıyla şifreleme, şifre çözme gibi işlemler yapılması isteniliyorsa, öncelikle cihazın işlemci hızı, bataryası, ram belleği gibi birçok donanımsal durumunu bilmemiz ve cihazımızın donanımsal yapısına uygun şifreleme algoritması ve anahtar boyutlarının seçilmesi gerekir. Bu durumun nedeni seçtiğimiz algoritma ve anahtar boyutuna göre anahtar oluşturma, şifreleme, şifre çözme gibi işlemlerinin daha az bir karmaşıklık ile sağlanabilmesinden kaynaklanmaktadır. Çünkü bir mobil cihaz için hız, performans artışı, bataryanın ömrü gibi durumlar çok önemlidir.

Yapılan örneklerle Eliptik eğri ve RSA sistemlerinin farklı anahtar boyutlarında anahtar oluşturma, şifreleme, şifre çözme, iletim işlemleri karşılaştırıldı. Bu karşılaştırma sonucunda Eliptik eğri altyapısı kullanılan şifreleme sisteminin RSA sistemine göre çok daha hızlı ve güvenilir olduğu sonucunu elde edildi.

RSA sisteminin Eliptik Eğri sistemine göre tek farkı daha büyük anahtar boyutlarıyla şifreleme ve şifre çözme işlemleri gerçekleştirmesidir ki bu durum daha uzun şifre çözme ve şifreleme işlem sürelerine neden olmaktadır. Fakat bu noktada uzun olarak kastedilen süreler bile milisaniyeler cinsinden olduğundan RSA algoritması ve anahtarları ile de kullanıcının farkında olmadan hızlı şifreleme ve şifre çözme işlemleri gerçekleştirilebilir.

RSA ve Eliptik Eğri sistemlerinin her ikisi de güvenilirdir, her ikisi de hızlıdır. Buradaki tek fark Eliptik Eğri sisteminin RSA'a göre çok daha hızlı olduğudur. Her iki algoritmada günümüz şifreleme işlemlerinde rahatlıkla kullanılabilir. Farklarının neler olduğu, hangi algoritmanın nasıl kullanılacağı Analiz ve Karşılaştırma kısmında açıklanmıştır.

İlerleyen zamanla birlikte sürekli gelişen teknoloji ile elde edilmek istenen amaçlardan birisi de kuantum bilgisayarlardır. Kuantum bilgisayarlarının çıkışı şu an için en güçlü sayılan RSA ve Eliptik Eğri sistemlerinden elde edilen ve çözülemez denilen şifrelerin çözülmesine neden olacağı açıktır. Kısacası kuantum bilgisayarların çıkışı günümüzde güvenilir kabul edilen bu algoritmaları da güvensiz hale getirilecek ve bu algoritmaları diğer birçok algoritma gibi emekliye ayıracaktır. Fakat en ilkel şifreleme yapılarından

en karmaşık şifreleme yapılarına kadar neredeyse bütün kriptolojik algoritmaların, zaman içerisinde kendinden önceki algoritmanın güvensizliğinden dolayı geliştirildiği unutulmamalıdır. Bir şeyi gizleme onu anlamsız hale getirme, anlamsız haldeki verileri çözmeye çalışma, hayatımıza girdiği andan bu yana gizemini korumaktadır. İlerleyen teknolojik gelişmelerin yanı sıra da farklı yapılara bürünerek gizemini korumaya devam edecektir.

5. KAYNAKLAR

[1] Rivest R. L., Shamir A., and Adleman L., A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Communications of the ACM*, February **(1978)**, 21(2):120-126.

[2] Miller V., Uses of Elliptic Curves in Cryptography, In H. C. Willams, Editor, *Advances in Cryptology: CRYPTO'85, volume 218 of Lectures Notes in Computer Science*, Springer-Verlag, **(1985)**, 417-426.

[3] Menezes A. J., Oorschot P. C. V., and Vanstone S. A., Handbook of Applied Cryptography, *CRC Press, Boca Raton, Florida, USA*, **(1997)**.

[4] Stinson D.R., Cryptography Theory and Practice, Second Edition, *Chapman & Hall/CRC*, CRC Press Company.

[5] Yavuz İ., Eliptik Eğri Kriptosisteminin FPGA Üzerinde Gerçeklenmesi, *Yüksek Lisans Tezi*, İstanbul Teknik üniversitesi, **(2008)**.

[6] http://www.muratyildirimoglu.com/makaleler/rsa_kripto_sistemi.htm (Erişim Tarihi : 12 Aralık **2015**).

[7] http://tr.wikipedia.org/wiki/DES (Erişim Tarihi : 12 Aralık **2015**).

[8] http://bilgisayarkavramlari.sadievrenseker.com/2009/06/03/aes-ve-rijndeal-sifreleme/ (Erişim Tarihi : 12 Aralık **2015**).

[9] http://tr.wikipedia.org/wiki/AES (Erişim Tarihi : 08 Ocak **2015**).

[10] http://tr.wikipedia.org/wiki/Açık_anahtarlı_şifreleme (Erişim Tarihi : 08 Ocak **2015**).

[11] Kodaz H., RSA Şifreleme Algoritmasının Uygulaması, *Akademik Bilişim 2003*, **(2013)**.

[12] Menezes A. J., Elliptic Curve Public Key Cryptosystems, *Kluwer Academic Publishers*, **(1993)**.

[13] Guajardo J., Paar C., Efficient Algorithms for Elliptic Curve Cryptosystems, Advances in Cryptology - *CRYPTO'97, LNCS 1294, 1997,* Springer-Verlag Berlin Heidelberg **(1997)**, pp.342-356.

[14] Neidhardt E., Asymmetric Cryptography for Mobile Devices.

[15] https://www.bilgiguvenligi.gov.tr/gizlilik/rsa-algoritmasi.html (Erişim Tarihi : 08 Ocak **2015**).

[16] http://www.daniellerch.me/doc/rsa-en.pdf (Erişim Tarihi : 08 Ocak **2015**).

[17] http://en.wikipedia.org/wiki/Lattice-based_cryptography (Erişim Tarihi : 08 Ocak **2015**).

[18] Kocher P. C., Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems, In *Advances in Cryptology—CRYPTO'96*, Springer Berlin Heidelberg, **(1996)**, pp. 104-113.

[19] Schindler W., A timing attack against RSA with the chinese remainder theorem, In *Cryptographic Hardware and Embedded Systems—CHES 2000*, Springer Berlin Heidelberg, **(2000)**, pp. 109-124.

[20] Pellegrini A., Bertacco V., Austin T., Fault-based attack of RSA authentication, In *Proceedings of the Conference on Design, Automation and Test in Europe*, European Design and Automation Association, **(2010)**, pp. 855-860.

[21] Genkin D., Shamir A., Tromer E., RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis, *IACR Cryptology ePrint Archive*, **(2013)**, 857.

[22] Ambedkar B. R., Gupta A., Gautam P., Bedi S. S., An Efficient Method to Factorize the RSA Public Key Encryption, *In Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, IEEE, **(2011)**, pp. 108-111.

[23] https://www.academia.edu/2428678/Eliptik_Eğri_Şifrelemesi (Erişim Tarihi : 08 Ocak **2015**).

[24] Yerlikaya T., Buluş E., Arda D., Eliptik Eğri Şifreleme Algoritması Kullanan Dijital İmza Uygulaması.

[25] Pollard J., Monte Carlo methods for index computation mod p, *Mathematics of Computation*, 32 **(1978)**, 918-924.

[26] http://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/D20-v2.pdf (Erişim Tarihi : 08 Ocak **2015**).

[27] Jansma N., Arrendondo B., Performance Comparison of Elliptic Curve and RSA Digital Signatures, *Technical Report from Sun Microsystems Laboratories*, **(2004)**.

[28] Koblitz N., Elliptic Curve Cryptosystem. *Mathematics of Computation*, **(1987)**, Vol. 48: 203-209.

[29] http://tr.wikipedia.org/wiki/Diffie-Hellman_anahtar_değişimi (Erişim Tarihi : 13 Mart **2015**).

[30] Diffie W., Hellman M.E., New directions in cryptography, *IEEE Transactions on Information Theory*, *IT-22*, **(1976)**, 644-654.

[31] Recommended Elliptic Curve Domain Parameters Daniel R.L.Brown

[32] http://tr.scribd.com/doc/203625259/Modified-Koblitz-Encoding-Method-for-ECC#scribd (Erişim Tarihi : 13 Mart **2015**).

[33] Hafizul Islam S. K., Biswas G. P., A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem, *Journal of Systems and Software* 84.11, **(2011)**, 1892-1898.

[34] Char A., Pierre E., Abdallah M., and Bachar E. H., A fast and secure elliptic curve based authenticated key agreement protocol for low power mobile communications, *Next Generation Mobile Applications, Services and Technologies,* **(2007)**, *NGMAST'07, The 2007 International Conference on*, IEEE.

[35] Ravikumar K., and Udhayakumar A., Secure Multiparty Electronic Payments Using ECC Algorithm: A Comparative Study, *Computing and Communication Technologies (WCCCT), 2014 World Congress on*, IEEE, **(2014)**.

[36] Saini R., and Kunwar S. V., Image Signcryption Using ECC, *Computational*

Intelligence and Communication Networks (CICN), 2014 International Conference on, IEEE, **(2014)**.

[37] Bakhtiari S., Baraani A., and Khayyambashi M.R., Mobicash: A new anonymous mobile payment system implemented by elliptic curve cryptography, *Computer Science and Information Engineering, 2009 WRI World Congress on. Vol. 3, IEEE,* **(2009)**.

[38] Lisonek D., and Drahansky M., Sms encryption for mobile communication, *Security Technology,* **(2008)**, *SECTECH'08, International Conference on*, IEEE.

[39] Somani U., Lakhani K., and Mundra M., Implementing digital signature with RSA encryption algorithm to enhance the Data Security of cloud in Cloud Computing, *Parallel Distributed and Grid Computing (PDGC), 2010 1st International Conference on*, IEEE, **(2010)**.

[40] Alrodhan W. A., Alturbaq A., and Aldahlawi S., A mobile biometric-based e-voting scheme, *Computer Applications & Research (WSCAR), 2014 World Symposium on*, IEEE, **(2014)**.

[41] Melgar M. E. V., Santander M., and Luz A., An alternative proposal of tracking products using digital signatures and QR codes, *Communications and Computing (COLCOM), 2014 IEEE Colombian Conference on, IEEE*, **(2014)**.

[42] Gani P. H., and Abdurohman M., Selective encryption of video MPEG use RSA algorithm, Information Technology, *Computer and Electrical Engineering (ICITACEE), 2014 1st International Conference on, IEEE,* **(2014)**.

[43] Agoyi M., and Seral D., SMS security: an asymmetric encryption approach, Wireless and Mobile Communications (ICWMC), *2010 6th International Conference on, IEEE,* **(2010)**.

[44] Saxena N., and Chaudhari S. N., Secure encryption with digital signature approach for Short Message Service, I*nformation and Communication Technologies (WICT),*

2012 World Congress on, IEEE, **(2012)**.

[45] Chen H. C., and Deviani R., A secure e-voting system based on rsa time-lock puzzle mechanism, *Proceedings of the 2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications, IEEE Computer Society,* **(2012)**.

[46] Figueroa K., Lopez E., and Garcia J. M., Electronic Voting System in Mexican Elections, *Computer Science (ENC), 2013 Mexican International Conference on, IEEE,* **(2013)**.

[47] Sahana A., and Misra I. S., Implementation of RSA security protocol for sensor network security: Design and network lifetime analysis, *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on, IEEE,* **(2011)**.

[48] Kodali R. K., Implementation of ECC with hidden Generator point in Wireless Sensor Networks, *Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on, IEEE,* **(2014)**.

[49] Zhao G., et al., RSA-based digital image encryption algorithm in wireless sensor Networks, *Signal Processing Systems (ICSPS), 2010 2nd International Conference on, Vol. 2, IEEE,* **(2010)**.

[50] Guptak S., An ethical way for image encryption using ECC, *Procofthe 1st International Conference on Computational Intelligence,* Communication Systemsand Networks 342 **(2009)**: 345.

[51] Wang C. T., Liao C. H., and Chen T. S., Audio-signal authenticating system based on asymmetric signature schemes, *Multimedia and Ubiquitous Engineering,* **(2007)**, *MUE'07, International Conference on, IEEE.*

[52] Rathanam G. J., and Sumalatha M. R., Dynamic secure storage system in cloud services, *Recent Trends in Information Technology (ICRTIT), 2014 International*

Conference on. IEEE, **(2014)** .

[53] http://www.msxlabs.org (Erişim Tarihi : 31 Haziran **2015**).

[54] http://yavuzbugra.wordpress.com (Erişim Tarihi : 31 Haziran **2015**).

[55] http://researchgate.net (Erişim Tarihi : 31 Haziran **2015**).

[56] http://www.bilgisayarkavramlari.com (Erişim Tarihi : 31 Haziran **2015**).

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : BODUR Hüseyin

Uyruğu : T.C.

Doğum tarihi ve yeri : 22.12.1989 / İzmir

Telefon : 0 (380) 542 10 36 / 4660

Faks : 0 (380) 542 10 37

E-posta : huseyinbodur@duzce.edu.tr

Eğitim

Derece	**Eğitim Birimi**	**Mezuniyet tarihi**
Lisans	Pamukkale Üniversitesi / Bilgisayar Müh.	2012
Lise	RASİM ÖNEL A.T.M.L	2007

İş Deneyimi

Yıl	**Yer**	**Görev**
20012-(devam ediyor)	Düzce Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce (KPDS : 78.75)

Yayınlar

1. Bodur H., Kara R., Zavrak S., RSA Şifreleme Algoritması Kullanılarak SMS İle Güvenli Mesajlaşma Yöntemi, *Akademik Bilişim 2015,* **(2015)**.

2.

3.